

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

STINGRAY GROUP INC. and STINGRAY MUSIC USA, INC.
Petitioners,

v.

EDWIN A. HERNANDEZ-MONDRAGON,
Patent Owner.

Case No. IPR2025-00350
U.S. Patent No. 10,524,002

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 10,524,002**

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	MANDATORY NOTICES PURSUANT TO 37 C.F.R. § 42.8(a)(1).....	1
	A. Real Parties-in-Interest (37 C.F.R. §42.8(b)(1))	1
	B. Related Matters (37 C.F.R. § 42.8(b)(2)).....	1
	C. Lead and Back-up Counsel (37 C.F.R. § 42.8(b)(3)).....	2
	D. Service Information (37 C.F.R. § 42.8(b)(4))	3
III.	STANDING UNDER 37 C.F.R. §42.104(a) & FEES	3
IV.	STATEMENT OF PRECISE RELIEF REQUESTED	3
	A. Claims for Which Review Is Requested (37 C.F.R. § 42.104 (b)(1))...3	
	B. Statutory Grounds of Challenge (37 C.F.R. § 42.104 (b)(2)).....	3
	C. Discretionary Denial is Not Warranted.....	4
V.	'002 PATENT SUMMARY	4
	A. Overview	4
	B. Prosecution History	8
	C. Effective Filing Date	11
VI.	SUMMARY OF THE PRIOR ART.....	11
	A. Avellan (Ex-1003).....	11
	B. Ma (Ex-1004)	14
	C. Fain (Ex-1005)	14
	D. Wannamaker (Ex-1006)	14
VII.	LEVEL OF ORDINARY SKILL IN THE ART.....	19
VIII.	CLAIM CONSTRUCTION (37 C.F.R. § 42.104(b)(3)).....	19
IX.	THE CHALLENGED CLAIMS ARE UNPATENTABLE.....	20
	A. Ground 1: Claims 1-6, 13 are obvious in view of Avellan.	20

a.	Claim 1	20
	1 pre... ..	20
	Element 1a	21
	Element 1b	24
	Element 1c	26
	Element 1d	26
	Element 1e	28
	Element 1f.....	29
	Element 1g.....	29
	Element 1h.....	31
b.	Claim 2	32
	Element 2	32
c.	Claim 3	33
	Element 3	33
d.	Claim 4.....	34
	Element 4 pre	34
	Element 4a	34
	Element 4b	34
	Element 4c	35
	Element 4d	35
	Element 4e	35
	Element 4f.....	35
	Element 4g.....	35
	Element 4h.....	36
e.	Claim 5	36
	Element 5	36
f.	Claim 6.....	36

	Element 6	36
g.	Claim 13	37
	Element 13 pre	37
	Element 13a	37
	Element 13b	37
	Element 13c	37
	Element 13d	37
	Element 13e	38
	Element 13f	38
	Element 13g	38
	Element 13h	38
B.	Ground 2: Claims 1-6, 13 are obvious in view of Avellan, Ma, and Fain.	40
a.	[1pre]-[1f]	40
b.	[1g]	40
c.	[1h]	42
d.	[2]	44
e.	[3]	46
f.	[4pre]-4[f]	47
g.	[4g]	47
h.	[4h], [5], [6]	48
i.	[13pre]-[13f]	49
j.	[13g]	49
k.	[13h]	49
C.	Ground 3: Claims 7-9 and 11-12 Are Anticipated by Wannamaker. .	49
a.	Claim 7	49

	Element 7 pre	49
	Element 7a	50
	Element 7b	53
	Element 7c	54
	Elements 7d & 7e.....	57
	Element 7f.....	59
	Element 7g.....	62
b.	Claim 8.....	63
	Element 8	63
c.	Claim 9.....	64
	Element 9	64
d.	Claim 11	66
	Element 11	66
e.	Claim 12.....	67
	Element 12	67
D.	Ground 4: Claim 10 Is Obvious in View of Wannamaker and Avellan.	70
a.	Claim 10.....	70
	Element 10	70
X.	CONCLUSION.....	74
XI.	CERTIFICATE OF WORD COUNT	76

TABLE OF AUTHORITIES

Cases

<i>Okajima v. Bourdeau</i> , 261 F.3d 1350 (Fed. Cir. 2001)	19
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005) (<i>en banc</i>)	19
<i>Toyota Motor Corp. v. Cellport Sys., Inc.</i> , IPR2015-00633, Paper 11 (PTAB Aug. 14, 2015).....	19

Statutes

35 U.S.C. § 102	1
35 U.S.C. § 102(a)(1).....	14
35 U.S.C. § 103	1
35 U.S.C. § 311	1, 3
35 U.S.C. § 314(a)	4
35 U.S.C. § 325(d)	4

Rules

37 C.F.R. § 42.10(b)	2
37 C.F.R. § 42.100(b)	19
37 C.F.R. § 42.104 (b)(1).....	3
37 C.F.R. § 42.104(a).....	3
37 C.F.R. § 42.104(b)(3).....	19
37 C.F.R. § 42.15(a).....	3
37 C.F.R. § 42.8(a)(1).....	1
37 C.F.R. § 42.8(b)(2).....	1
37 C.F.R. § 42.8(b)(3).....	2
37 C.F.R. § 42.8(b)(4).....	3

PETITIONERS' EXHIBIT LIST

Exhibit	Short name	Description
Ex-1001	'002 Patent	U.S. Patent No. 10,524,002
Ex-1002	'002 File History	File History of U.S. Patent No. 10,524,002
Ex-1003	Avellan	U.S. Patent No. 8,954,600
Ex-1004	Ma	U.S. Patent Application Publication No. 2014/0150019
Ex-1005	Fain	"Java Programming 24-Hour Trainer", Yakov Fain
Ex-1006	Wannamaker	U.S. Patent Application Publication No. 2004/0031052
Ex-1007	Lipoff Decl.	Declaration of Stuart Lipoff including <i>Curriculum Vitae</i>
Ex-1008	Munford Decl.	Declaration of June Munford including <i>Curriculum Vitae</i>
Ex-1009	'074 Patent	U.S. Patent No. 10,123,074
Ex-1010	'074 File History	File History of U.S. Patent No. 10,123,074
Ex-1011	Luthra	U.S. Patent Application Publication No. 2014/0068690
Ex-1012	SAC	Second Amended Complaint from related litigation, Case No. 1:24-cv-21226
Ex-1013	Sarginson	P.A. Sarginson, "MPEG-2: a tutorial introduction to the systems layer," IEE Colloquium on MPEG-2 – What it is and What it isn't, London, UK, 1995, pp. 4/1-4/13

LISTING OF CHALLENGED CLAIMS¹

Element	Claim Language
Claim 1	
[1 pre]	A computer-implemented method, comprising:
[1a]	receiving, from a content provider, a request for at least one media stream for playback on a broadcast media channel, wherein the at least one media stream includes a plurality of multimedia items of different types;
[1b]	obtaining content corresponding to the plurality of multimedia items from at least one source offering the content in at least one first format;
[1c]	rendering a web page by a browser using the content;
[1d]	generating a temporal sequence of screen captures of the rendered web page, where each screen capture defines all the content of the web page at a given time, and at least two adjacent screen captures illustrate a dynamic change of at least a portion of the content over time;
[1e]	assembling the at least one media stream using the temporal sequence of screen captures; and
[1f]	providing the at least one media stream to the content provider for broadcast on the broadcast media channel;
[1g]	wherein the at least one media stream corresponds to at least one of an HTTP live stream (HLS), an HTTP playlist, and a Real-time Streaming Transport (RTSP) stream; and
[1h]	wherein the web page is rendered in parallel in multiple threads.
Claim 2	

¹ As the Petition presents the claim listing in its body in fragments and/or grouped by limitation type, rather than in numerical order, all claim limitations are presented here together so the Panel may easily view them in order and in context.

Element	Claim Language
[2]	The method according to claim 1, wherein the at least one media stream is interacted with MPEG two way control messages.
Claim 3	
[3]	The method according to claim 1, wherein MPEG messages are sent to a service provider via a unicast address.
Claim 4	
[4 pre]	A computer-implemented method, comprising:
[4a]	receiving, from a content provider, a request for at least one media stream for playback on a broadcast media channel, wherein the at least one media stream includes a plurality of multimedia items of different types;
[4b]	obtaining content corresponding to the plurality of multimedia items from at least one source offering the content in at least one first format;
[4c]	rendering a web page by a browser using the content;
[4d]	generating a temporal sequence of screen captures of the rendered web page, where each screen capture defines all the content of the web page at a given time, and at least two adjacent screen captures illustrate a dynamic change of at least a portion of the content over time;
[4e]	assembling the at least one media stream using the temporal sequence of screen captures; and
[4f]	providing the at least one media stream to the content provider for broadcast on the broadcast media channel;
[4g]	wherein assembling the at least one media stream comprises inserting at least one MPEG packet including metadata corresponding to the multimedia item; and
[4h]	wherein the web page is rendered in parallel in multiple threads.

Element	Claim Language
Claim 5	
[5]	The computer-implemented method of claim 4, wherein the at least one media stream is interacted with MPEG two way control messages.
Claim 6	
[6]	The computer-implemented method of claim 4, wherein MPEG messages are sent to a service provider via a unicast address.
Claim 7	
[7 pre]	A system, comprising:
[7a]	a multicast server configured to generate parameters relating to at least one of video content, image content and audio content, and configured to output the parameters via MPEG outputs;
[7b]	a caching server communicatively coupled to the multicast server,
[7c]	and configured to receive and store the parameters,
[7d]	create a temporal sequence of screen captures of a rendered web page, where each said screen capture defines content of the rendered web page at a given time, and at least two adjacent screen captures in the temporal sequence illustrate a dynamic change of at least a portion of the video content, image content or audio content over time,
[7e]	assemble the temporal sequence of screen captures, and
[7f]	provide at least one of the video content, image content and audio content to the multicast server for generating a multicast stream with MPEG encoded video and audio; and
[7g]	a monitoring system configured to maintain generation of video and audio files, even in failure cases.
Claim 8	
[8]	The system according to claim 7, further comprising multiple

Element	Claim Language
	servers for load balancing and fault-tolerance.
Claim 9	
[9]	The system according to claim 7, wherein the multicast server is selected to communicate a multicast stream available for a broadband network.
Claim 10	
[10]	The system according to claim 7, wherein the video content is encoded in a video streaming format so as to generate a unicast stream.
Claim 11	
[11]	The system according to claim 7, wherein the system generates audio MPEG files and add [sic] metadata in an MPEG frame.
Claim 12	
[12]	The system according to claim 7, wherein the system generates MPEG metadata that is interacted with two way broadcast TV systems.
Claim 13	
[13 pre]	A computer-implemented method, comprising:
[13a]	receiving, from a content provider, a request for at least one media stream for playback on a broadcast media channel, wherein the at least one media stream includes a plurality of multimedia items of different types;
[13b]	obtaining content corresponding to the plurality of multimedia items from at least one source offering the content in at least one first format;
[13c]	rendering a web page by a browser using the content;
[13d]	generating a temporal sequence of screen captures of the rendered web page, where each screen capture defines all the content of the

Element	Claim Language
	web page at a given time, and at least two adjacent screen captures illustrate a dynamic change of at least a portion of the content over time;
[13e]	assembling the at least one media stream using the temporal sequence of screen captures; and
[13f]	providing the at least one media stream to the content provider for broadcast on the broadcast media channel;
[13g]	wherein the at least one media stream corresponds to at least one of an HTTP live stream (HLS), an HTTP playlist, and a Real-time Streaming Transport (RTSP) stream; and
[13h]	wherein the web page is rendered in parallel in virtual instances or virtual machines.

I. INTRODUCTION

Under 35 U.S.C. § 311, Petitioners Stingray Group Inc. and Stingray Music USA, Inc. request *inter partes* review (“IPR”) of claims 1-13 (“Challenged Claims”) of U.S. Patent No. 10,524,002 (“the ’002 Patent” or “Patent-at-Issue”). Petitioner requests institution of this Petition and a finding that the Challenged Claims are unpatentable under (post-AIA) 35 U.S.C. §§ 102 and 103.

II. MANDATORY NOTICES PURSUANT TO 37 C.F.R. § 42.8(a)(1)

A. Real Parties-in-Interest (37 C.F.R. §42.8(b)(1))

Stingray Group Inc. and its subsidiary Stingray Music USA, Inc. (collectively “Stingray” or “Petitioners”) are the real parties-in-interest. No other party has directed, funded, or controlled filing of this Petition, and this Petition was not filed at the behest of any other party.

B. Related Matters (37 C.F.R. § 42.8(b)(2))

The Patent-at-Issue is currently the subject of pending litigation in *Hernandez, et al. v. Stingray Group Inc., et al.*, Case No. 1:24-cv-21226 (S.D. Fla. Apr. 2, 2024). Edwin A. Hernandez-Mondragon is the current assignee of record for the Patent-at-Issue. In the related litigation, Edwin A. Hernandez-Mondragon has represented that he “is the owner of all rights, title and interests to the ’002 Patent.” Ex-1012, ¶ 160. In the same litigation, he also represents that his

company, EGLA Corp., “has an exclusive license to the technologies owned by Dr. Hernandez.” *Id.*, ¶ 20.

The parent of the Patent-at-Issue is U.S. 10,123,074 (“the ’074 Patent”, Ex-1009). The ’074 Patent is the parent to both U.S. Patent No. 10,524,002 and U.S. Patent No. 11,140,441. Petitioners are currently challenging all claims of the ’074 Patent in IPR2025-00349 and all claims of U.S. Patent No. 11,140,441 in IPR2025-00351.

C. Lead and Back-up Counsel (37 C.F.R. § 42.8(b)(3))

As shown in the Power of Attorney filed concurrently herewith under 37 C.F.R. §42.10(b), Petitioners designate lead and back-up counsel as follows:

Lead Counsel	Backup Counsel
Masood Anjom Reg. No. 62,167 ALAVI & ANAIPAKOS PLLC 609 Main Street, Suite 3200 Houston, Texas 77002 Phone: (713) 751-2367 Facsimile: (713) 751-2341 manjom@aatriallaw.com	Joshua S. Wyde Reg. No. 57,698 ALAVI & ANAIPAKOS PLLC 609 Main Street, Suite 3200 Houston, Texas 77002 Phone: (713) 751-2365 Facsimile: (713) 751-2341 jwyde@aatriallaw.com

D. Service Information (37 C.F.R. § 42.8(b)(4))

Petitioners may be served paper copies as specified above. Petitioners consents to electronic service at the following email addresses: manjom@aatriallaw.com, jwyde@aatriallaw.com, and IPR2025-00350@aatriallaw.com.

III. STANDING (37 C.F.R. §42.104(a)) & FEES

Petitioners certify that the '002 Patent is eligible for *IPR* and Petitioners are not barred or estopped from requesting an *IPR* on the grounds identified herein.

The office is authorized to charge the fee set forth in 37 C.F.R. § 42.15(a) and any additional fees that might be due in connection with this petition to Deposit Account No. 604950.

IV. STATEMENT OF PRECISE RELIEF REQUESTED**A. Claims for Which Review Is Requested (37 C.F.R. § 42.104 (b)(1))**

Petitioner requests *inter partes* review and cancellation of Challenged Claims 1–13 of the Patent-at-Issue under 35 U.S.C. § 311.

B. Statutory Grounds of Challenge (37 C.F.R. § 42.104 (b)(2))

Ground	Prior Art References	Basis	Claim(s) Challenged
1	Avellan	§103	1-6, 13
2	Avellan, Ma and Fain	§103	1-6, 13
3	Wannamaker	§102	7-9, 11-12
4	Wannamaker and Avellan	§103	10

C. Discretionary Denial is Not Warranted

Petitioners respectfully submit that the Board should not exercise its discretion under 35 U.S.C. §§ 314(a) or 325(d) to deny this Petition. Avellan, Fain, and Wannamaker were not considered during prosecution and explicitly disclose elements found to be missing during prosecution. Ma was the basis of rejection for certain limitations during prosecution which the applicant did not traverse. Additionally, the related district court case is still at an early stage and Petitioners intend to move for a stay of that litigation pending this Petition. Therefore, discretionary denial is not warranted.

V. '002 PATENT SUMMARY

A. Overview

The Patent-at-Issue is generally directed to obtaining media content from a cloud-based system (e.g., a website) and assembling the content into a media stream for delivery to a content provider such as a cable or satellite operator. Ex-1001, [Abstract], 1:19-21. The Patent-at-Issue states:

An example computer-implemented method can receive, from a content provider such as a cable or satellite operator, a request for at least one media stream for playback on a broadcast media channel. The request for the at least one media stream can include a request for a plurality of multimedia items. Content corresponding to the plurality of multimedia items can be obtained from at least one website or cloud service that offers the content in at least one

first format. Based on the obtained content, the plurality of multimedia files can be generated in a second format that is compatible with the content provider. The media stream can [be] assembled using each of the multimedia items generated in the second format. The media stream can be provided to the content provider for broadcast on the broadcast media channel.

Ex-1001, 2:53-67.

The Patent-at-Issue discloses a Mediaplug 110 unit that is in communication with cloud 101 through the internet 105. Ex-1001, 3:63-4:17; Fig. 1. The Mediaplug can reside at an “MSO” facility “used by a cable operator to receive all TV and Radio programming that the cable operator will then multiplex, re-encode, and distribute from the MSP to a set of subscriber units with set top boxes....” *Id.*, 4:17-24. The Mediaplug 110 provides multimedia content (e.g., video and audio) to an MPEG multiplexer which delivers content to subscriber set top boxes 130, 131, 132 via coaxial cable 125. *Id.*, 4:25-33.

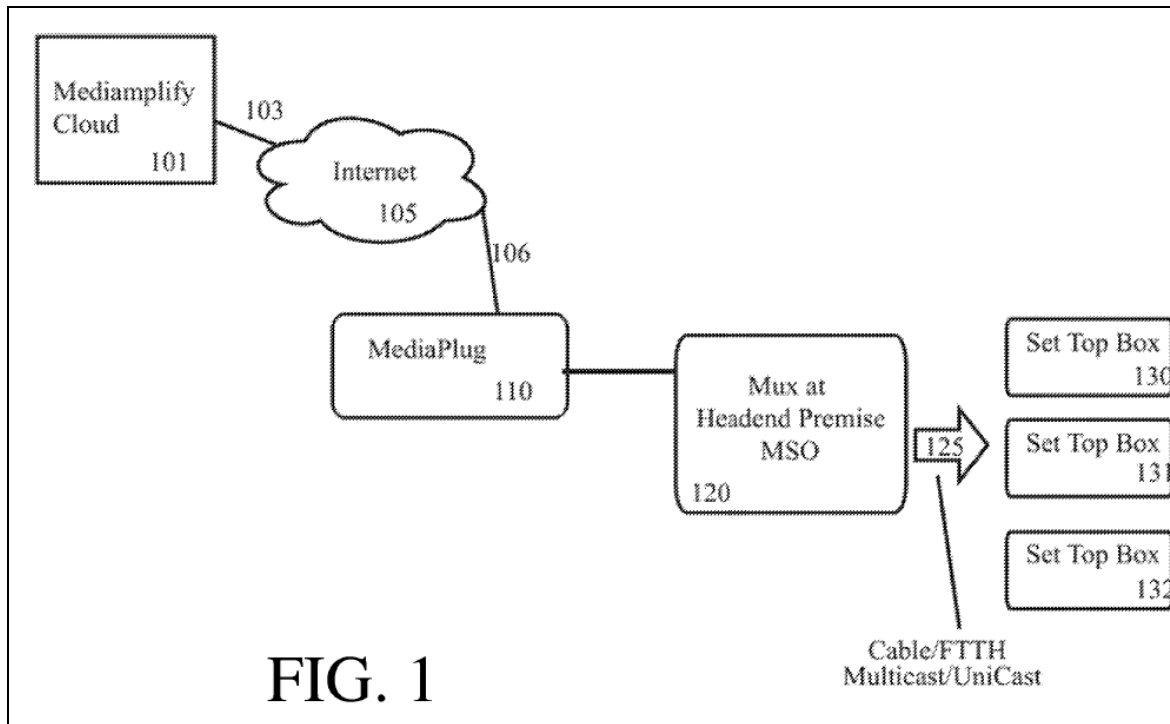
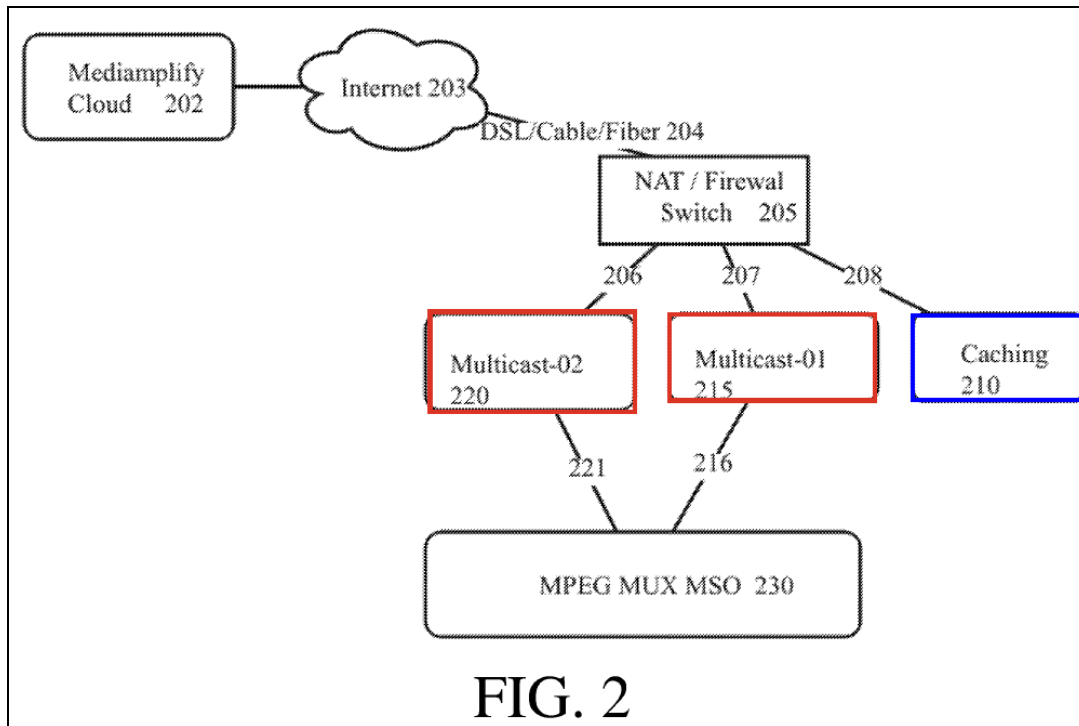


Fig. 2 shows the hardware components of the Mediaplug 110 including three servers. Ex-1001, 4:49-51. Specifically, the Mediaplug 110 includes a caching unit 210 (blue) which is the system's storage component and can contain the multimedia files in the format required by the cable operators. *Id.*, 4:66-5:2. The caching unit can encode and transcode the multimedia files it downloads from the cloud and communicates with the two multicast units 215, 220 (red) allowing the multicast units to pull or obtain information from it. *Id.*, 5:3-9.



As discussed further below, the following two limitations were deemed allowable during prosecution: (1) “wherein the webpage is rendered in parallel in multiple threads”; and (2) “a monitoring system configured to maintain generation of video and audio files, even in failure cases.” Ex-1002, 35-44, 96-97. These limitations were incorporated in the independent claims to place them in condition for allowance. *Id.*, 27-34.

With respect to the term “thread” of first limitation, the specification states: “The caching unit encodes audio and video, and can use the disk space in the Mediaplugin. Hence in 508 the method checks for sufficient space left in the memory or hard drive. This task can be run on a thread together with all the other

tasks that are working in parallel in order to speed up the processes of encoding and transcoding.” Ex-1001, 6:43-48 (emphasis added²).

With respect to the second limitation, the specification discloses (1) “Monit 750 corresponds to a monitoring block that can ensure that the process running the ‘Media Parallel encoder’ keeps running at all times and does not stop upon failure.” *Id.*, 8:63-66; (2) “the multicast-01 and multicast-02 blocks contain a ‘Recording’ function 790, which is in charge or (*sic.*) recording the multicasting streams and recording for later use in case of failure on the system.” *Id.*, 9:55-58; (3) “... the streamer can perform functions that include: connect to the caching unit; retrieve generated videos per playlist; determine if unit is working in fault-tolerant mode; playback pre-recorded audio if a failure is detected...” *Id.*, 9:66-10:4; and (4) “In the case that there are no ping responses, a timeout will trigger a failure and a pre-recorded content 820 will be played or sent to the multicast address and port of record.” *Id.*, 10:42-45.

B. Prosecution History

During prosecution of the parent ’074 Patent the Examiner rejected the original independent claims as anticipated by the cited prior art. Ex-1010, 74-92. In response, the Patent Owner amended the claims to require: (1) “render[ing] a web page by a browser using the content,” and (2) “generat[ing] a temporal sequence of

² Unless otherwise stated, all emphases are added.

screen captures of the rendered webpage, where each screen capture defines all the content of the web page at a given time, and at least two adjacent screen captures illustrate a dynamic change of at least a portion of the content over time.” Ex-1010, 62. Based on the applicant’s amendment and representation that this addition was “an important distinction” over the cited prior art (Ex-1010, 69), the Examiner allowed the claims without further rejection. Ex-1010, 21-23.

With respect to the added “rendering” language, the Patent-at-Issue explains that a web service “can include JavaScript, CSS, and HTML content files that can be rendered using a browser using WebKit (e.g. PhantomJS or Safari) or any other browser-rendering engine compatible with HTML4/5 or any future HTML version.” Ex-1001, 7:2-6.

With respect to the added “screen capture” language, the Patent-at-Issue states “rendered images from a web service can be used to create overlapping video or replace any video in the video feed file.” *Id.*, 7:7-9. Specifically, “one or more screens will be retrieved from the rendering engine” and captured “using a tool similar to PhantomJS (Webkit-based headless engine for rendering images) or QtCapture, or simply capturing screen frame buffers from a machine running any software tool.” *Id.*, 7:16-23. “Once all screens and video are captured, 530 the resulting output created can be a video file that then is either replaced or merged (multiplexed) with the original file.” *Id.*, 7:26-29. “The multiplexing can be done

using tools such as FFMPEG or VLC or any proprietary tool that can multiplex audio and video files into a resulting MPEG Transport Stream (or MPEG-4 output, or any similar output).” *Id.*, 7:41-45; 7:34-36.

Accordingly, the Patent-at-Issue explicitly discloses that both the “rendering” and “screen capture” limitations can be performed using commonly-known and readily-available third-party software tools, such as PhantomJS and/or QtCapture. Ex-1007, ¶86. Similarly, the screen captures generated can be multiplexed using commonly-known and readily-available third-party tools like FFMPEG or VLC. Ex-1007, ¶89. Indeed, even the resulting MPEG Transport Stream output format was commonly known and used prior to the Patent-at-Issue’s filing date. Ex-1007, ¶94.

The independent claims of the Patent-at-Issue generally mirror the claims of the parent ’074 Patent with added limitations that were deemed allowable. Specifically, during prosecution, applicant filed a preliminary amendment replacing claims 1-23 with new claims 24-38. Ex-1002, 95-97. Except for dependent claims 26, 30 and 36 which were deemed allowable, the remaining claims were rejected on grounds of double-patenting in view of the ’074 Patent alone or in combination with other references. Ex-1002, 37-43. Claims 26 and 30 added the “multiple threads” limitation (i.e., “wherein the web page is rendered in parallel multiple threads”) and claim 36 added the “monitoring system” limitation

(i.e., “a monitoring system configured to maintain generation of video and audio files, even in failure cases”). Ex-1002, 96-97. In response, applicant did not traverse the rejections and instead, amended independent claims 24, 28 and 39 to incorporate the “multiple threads” limitation and amended independent claim 32 to incorporate the “monitoring system” limitation of allowable claims. Ex-1002, 28-31. A notice of allowance then issued. Ex-1002, 7-14.

As discussed below, the use of multiple threads and a monitoring system was commonly known prior to the filing date of the Patent-at-Issue. Ex-1007, ¶97. Therefore, the Challenged Claims are all directed to an assemblage of pre-existing commonly-known processes, software tools and transport formats. Ex-1007, ¶100.

C. Effective Filing Date

The Patent-at-Issue was filed on October 5, 2018, claiming priority to a provisional application filed December 22, 2014. Ex-1001, (60). Petitioners assume a priority date of December 22, 2014, without conceding that any of the Challenged Claims are actually entitled to that priority.

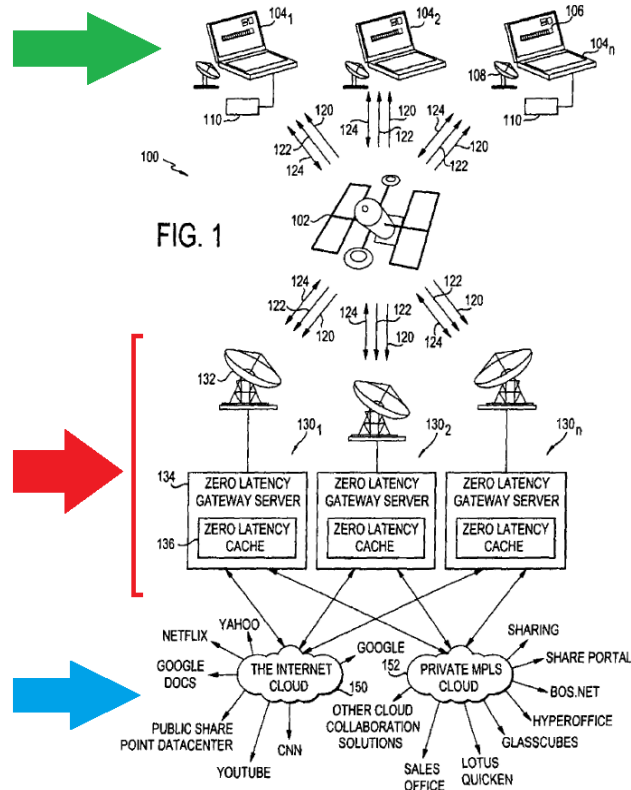
VI. SUMMARY OF THE PRIOR ART

A. Avellan (Ex-1003)

Avellan is U.S. Patent No. 8,954,600 to Avellan, et al., filed on March 2, 2012, and issued on February 10, 2015, making it prior art under at least 35 U.S.C. §102(a)(2).

Avellan discloses a communication system that can “broadcast information from a web server over a broadcast device to a multitude of user computers.” Ex-1003, 2:9-10. “[S]ystem 100 provides for the user computers 104₁-104_n to access the Internet 150 and/or private network cloud 152 through a virtual browser 106 of the gateway 130.” Ex-1003, 4:18-21. “[T]he browser software 106 resides at the gateway 134, and is configured to run as a virtual browser as a remote client of the gateway server 134.” Ex-1003, 4:56-58. Because the browser software resides at the gateway, “the remote browser 106 may rely upon the computing power and very high-speed connection of the gateway server 134 to resolve and render pages, images, and documents from the networks 150, 152.” Ex-1003, 4:62-65.

Figure 1 (below) shows a user at computer 104 (green) can request media streams from either the Internet or a private network cloud (collectively, blue) for playback. The gateway platform 130 (red) receives the request, obtains the content from the networks, processes it using the disclosed browser software, and then broadcasts the requested media on a broadcast media channel.



Ex-1003, Fig. 1.

Avellan adds that “web pages often have information which is rendered over a period of time, *such as a video*, streaming information, or an animated display. Accordingly, the gateway server 134 *continuously captures images of the web page* until it detects a repeating pattern of the information in the web page.” Ex-1003, 5:5-5:10. “Thus, gateway server 134 will be certain to capture all the information on the web page even if that information continues over a period of time.” *Id.*, 5:10-5:12. Avellan explicitly states that “suitable sites for the broadcast mode are news sites like CNN, BBC, newspaper, *live streaming television feeds*, *movies*, music downloads, or ESPN.” *Id.*, 8:18-21.

B. Ma (Ex-1004)

Ma is U.S. Patent Application Publication No. 2014/0150019 to Ma, et al., published on May 29, 2014, making it prior art under at least 35 U.S.C. §102(a)(1).

Ma generally relates to over-the-top (OTT) media delivery for targeted advertisement replacement for near-real-time streaming media. Ex-1004, [0001].

C. Fain (Ex-1005)

Fain is “Java Programming 24-Hour Trainer”, Yakov Fain, published by Wiley Publishing, Inc. (“Fain”) and publicly available at least as early as 2011. Ex-1008. Fain is prior art under at least 35 U.S.C. §102(a)(1).

Fain is a training book “for anyone who wants to learn how to program with the Java language” and includes a chapter titled “Introduction to Multi-Threading.” Ex-1005, xxi, xxix.

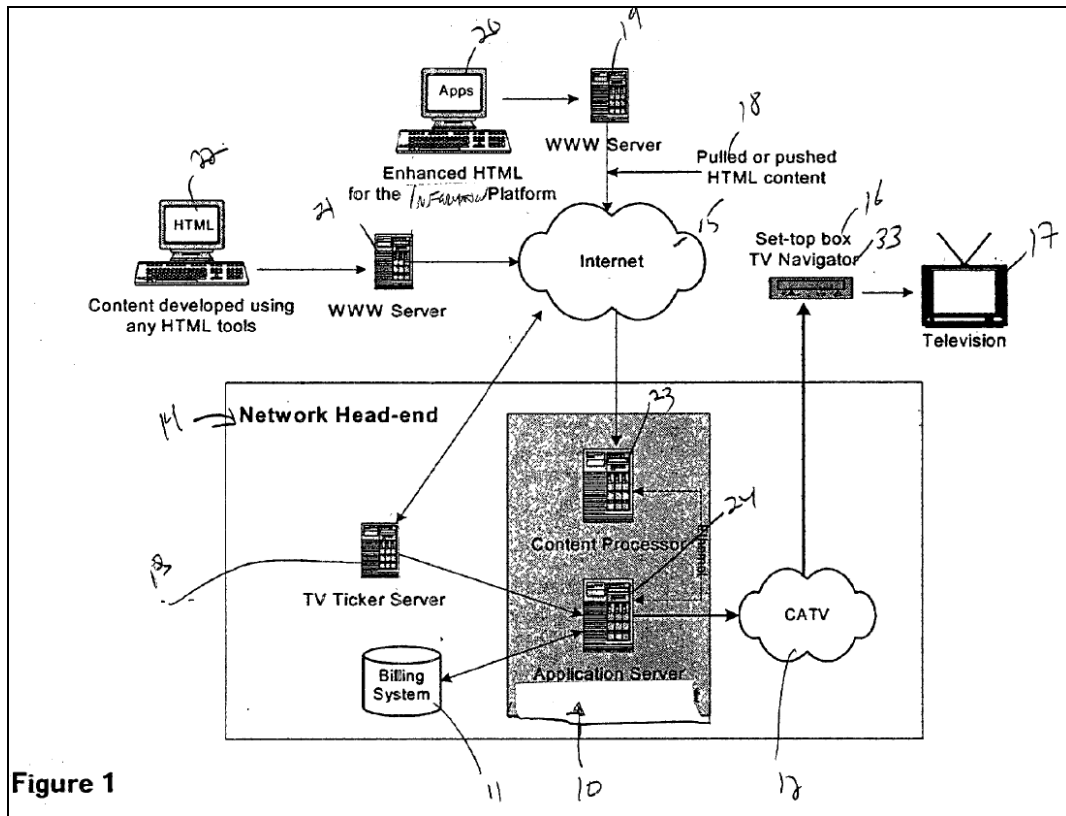
D. Wannamaker (Ex-1006)

Wannamaker is U.S. Patent Application Publication No. 2004/0031052 to Wannamaker, et al., published on February 12, 2004, making it prior art under at least 35 U.S.C. §102(a)(1).

Wannamaker discloses a “middleware solution” called an “information platform” which “provides a platform for cable system operators to deliver a full range of sophisticated new products to cable television subscribers.” Ex-1006 [Abstract]. “The information platform mediates the communication between []

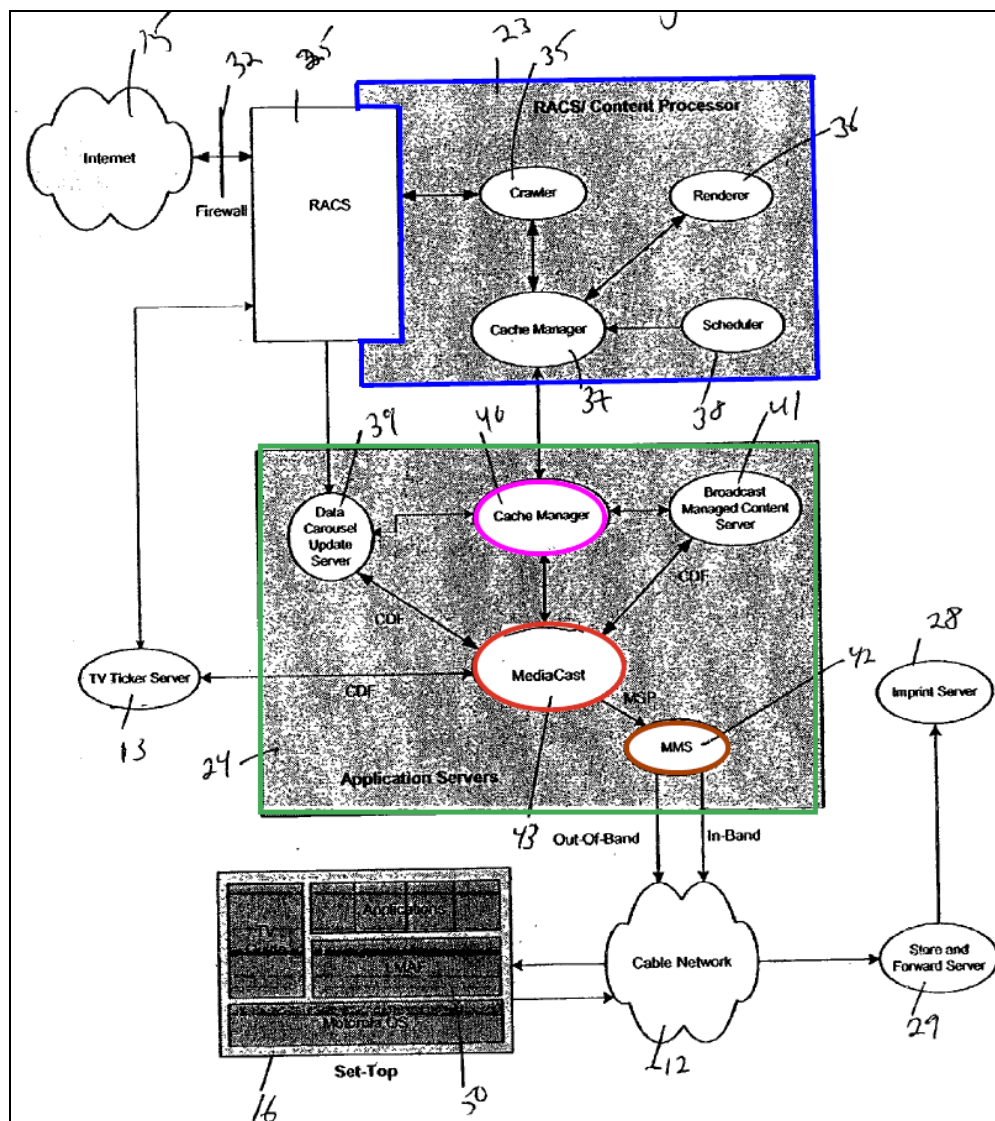
Web-based content and the television set” by “convert[ing] Internet content and protocols into forms suitable for TV viewing” and “supports HTML 3.2, including text, tables, forms, images, animated images, image maps, frames, and background images.” Ex-1006, [0098], [0136], [Abstract].

Fig. 1 (below) shows the information platform “consists of application servers 10 installed at the network head end 14 and hosted on a Unix platform, and the TV navigator client hosted on a set-top box 16.” Ex-1006, [0058]. The system provides “proprietary applications, third-party applications, and managed broadcast content to subscribers, along with broadcast television.” *Id.* “The application server transcodes Web pages into a format that is viewable on the set-top box.” *Id.*, [0132].



The application server allows “exchange of information with a CATV system 12 and the Internet 15.” Ex-1006, [0059]. “The CATV system provides communications between the application server and the subscriber set-top box 16 on which a TV navigator is running 33 and to which the subscriber’s television 17 is connected.” *Id.*, [0059]. The application server is also connected (via the internet) “to one or more WWW servers 19, 21, which support systems that supply enhanced HTML 20 to provide pushed or pulled HTML content 18 to subscribers; and systems 22 which supply content that is developed using any HTML tools.” *Id.*, [0059].

Fig. 3 (below) shows the “process of collecting, caching, transcoding, and delivering content” with the arrows indicating data flows. Ex-1006, [0099]. The application server 10 is comprised of an application server processor 24 (green) and a content processor 23 (blue). *Id.*, [0059], [0064]. The content processor 23 retrieves and transcodes HTML broadcast managed content into I-frames, and data to be displayed by the microbrowser on the set-top box 16. *Id.*, [0068].



The application server (24) hosts a variety of processes including the host monitoring server, the Information Mediacast (data carousel server and multimedia streamer), and the broadcast managed content server. *Id.*, [0069]-[0076]. The information Mediacast server manages requests to transmit the carousels of data and image files and is composed of the data carousel server (red) for data formatting, and Multimedia Streamer (MMS) components (brown) for data transmission. *Id.*, [0169]. The MMS “consists of the audio/video/terminal server (AVTS) that delivers the units of audiovisual information sent by the system to the user’s set-top box.” *Id.*, [0193]. Specifically, “[t]he AVTS is responsible for delivering MPEG video and Dolby AC-3 audio streams to the set-top box” using an MPEG multiplexer (MUX) card. *Id.*, [0195].

The content processor 23 “produces digital cable system compatible (rendered) content from standard Web content retrieved from the Internet” and cache managers (37, 40) support “smart caching” operations between the application server 24 and the content processor 23. *Id.*, [0235]-[0244].

Wannamaker also discloses a microbrowser which uses a thin client model to provide a web browser function on the digital set-top boxes by continuously transmitting web data to the client in an MPEG video stream. Ex-1006, [0267]-[0273].

Avellan, Ma, Fane and Wannamaker are in the same field of endeavor as the Patent-at-Issue. Ex-1007, ¶105, 108, 111, 120.

VII. LEVEL OF ORDINARY SKILL IN THE ART

The level of ordinary skill in the art may be reflected by the prior art of record. *See Okajima v. Bourdeau*, 261 F.3d 1350, 1355 (Fed. Cir. 2001). Here, a Person of Ordinary Skill in the Art (“POSITA”) at the time of the claimed invention would have a bachelor’s or equivalent degree in computer science or a related field and at least 2 years of experience in multimedia processing or delivery using internet protocols. Ex-1007, ¶68. A person with less formal education but more experience, or more formal education but less experience, could also meet the standard. Ex-1007, ¶69.

VIII. CLAIM CONSTRUCTION (37 C.F.R. § 42.104(b)(3))

Claim terms in an IPR are construed according to their “ordinary and customary meaning” as would have been understood by a POSITA, at the time of the invention, having taken into consideration the language of the claims, the specification, and the prosecution history of record. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1313 (Fed. Cir. 2005) (*en banc*); 37 C.F.R. § 42.100(b). The Board, however, only construes the claims when necessary to resolve the underlying controversy. *Toyota Motor Corp. v. Cellport Sys., Inc.*, IPR2015-00633, Paper 11 at 16 (PTAB Aug. 14, 2015). Petitioners respectfully submit that the Board need

not expressly construe any terms because the prior art invalidates the Challenged Claims under any plausible construction.

IX. THE CHALLENGED CLAIMS ARE UNPATENTABLE

A. Ground 1: Claims 1-6, 13 are obvious in view of Avellan.

a. Claim 1

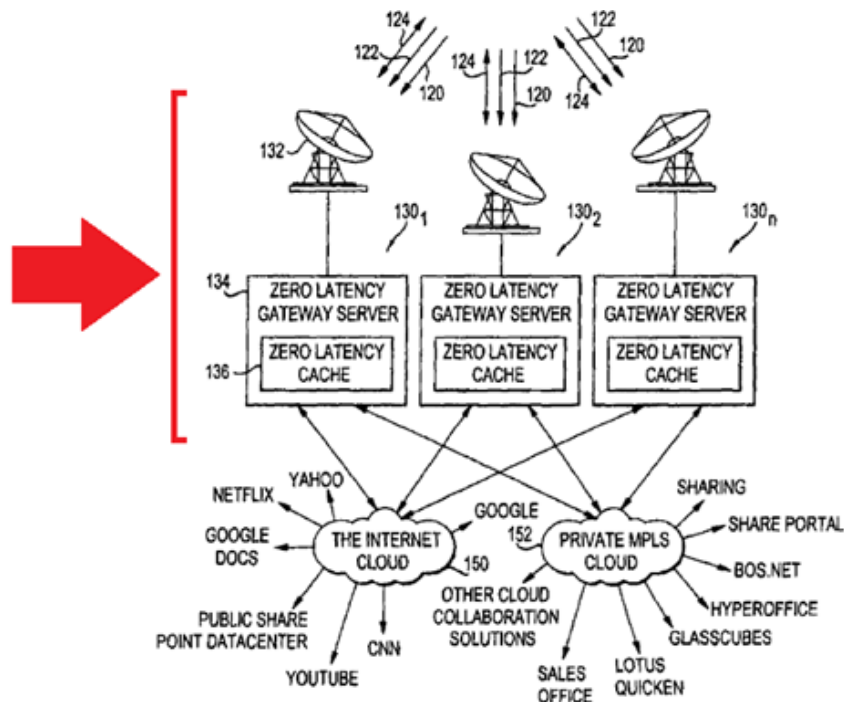
1 pre

[1 pre]

A computer-implemented method comprising:

To the extent the preamble is limiting, Avellan discloses it.

Avellan discloses gateway platforms 130 (red), each of which “includes a respective satellite antenna 132 or a broadcast transmitter/receiver, and a gateway server 134 [and] a cache 136.” Ex-1003, 3:64-66.



“The gateway server 134 can be a server farm that is adapted to act as a secure and fast gateway to the Internet 150, 152 and may be a network of servers, mainframe computer(s) or other high speed and high capacity processing devices.”

Ex-1003, 4:26-29.

Element 1a

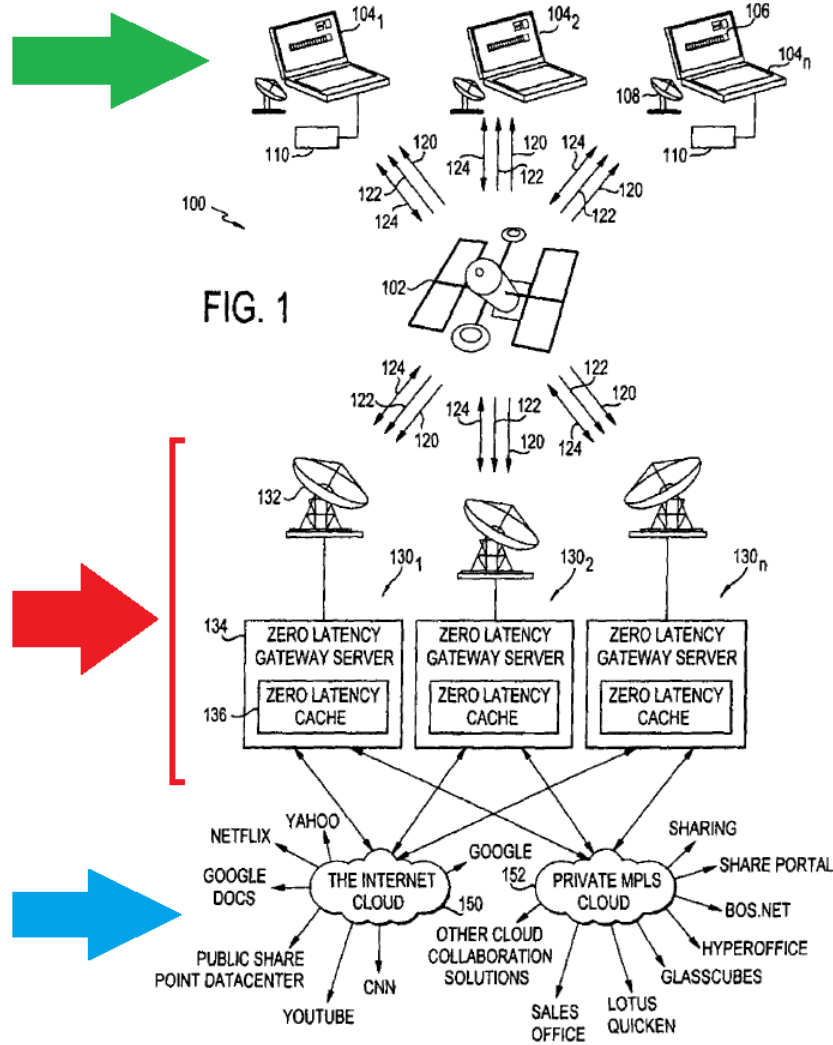
[1a]	receiving, from a content provider, a request for at least one media stream for playback on a broadcast media channel, wherein the at least one media stream includes a plurality of multimedia items of different types;
-------------	---

Avellan teaches this element.

First, Avellan’s “system 100 provides for the user computers 104₁-104_n to access the Internet 150 and/or private network cloud 152 through a virtual browser 106 of the gateway 130.” Ex-1003, 4:18-21. “The user operates the browser 106 which sends a signal over the satellite 102 to the gateway 130,” which in turn “accesses the networks 150, 152 and returns the results to the virtual browser 106 via the satellite 102.” *Id.*, 4:21-25; 6:22-25 (browser 106 “sends commands and a tag file to the gateway server 134 so that the gateway server 134 retrieves the information desired by the user”). “[T]he browser software 106 resides at the gateway 134, and is configured to run as a virtual browser as a remote client of the gateway server 134.” *Id.*, 4:56-58.

“As shown in [Avellan’s] Fig. 1, the broadcast device 102 can be a satellite 102 which communicates with a satellite antenna 132 at the gateway end, and also communicates with a repeater or the link (not shown) at the user end. The repeater in turn communicates with a plurality of user computers 104 over a wired or wireless connection 108.” Ex-1003, 4:36-41. “[T]he gateway platform 130 can broadcast information over the antennas 132 and the broadcast device 102 to a large number of user computers 104 via the wired or wireless connection 108.” *Id.* 4:46-49.

Therefore, referring to Avellan’s Figure 1 (below), Avellan discloses that a user at user computer 104 (green) can request media streams from either the Internet or a private network cloud (collectively, blue) for playback. That request is received by gateway platform 130 (red), which obtains the content from the networks and processes it using the disclosed browser software, and then broadcasts the requested media on a broadcast media channel. Therefore, a request for at least one media stream is received from a content provider, the gateway platform 130.



Ex-1003, Fig. 1.

Second, Avellan discloses that the media stream includes a plurality of multimedia items of different types noting that “there can be a number of broadcast channels and each channel can have a specific theme, such as country music, military television, etc.” Ex-1003, 8:25-27. “The user can then tune the browser 106 to one of the broadcast channels to receive that content. Thus, the broadcast channel can be configured here to receive live content, or to update the storage

device 110 of the user computer 104.” *Id.*, 8:27-31. For instance, “[i]f a large number of viewers choose to select a certain hockey game, that page of the website can also be sent in the broadcast mode.” *Id.*, 8:34-36. Avellan teaches that “suitable sites for the broadcast mode are news sites like CNN, BBC, newspaper, live streaming television feeds, movies, music downloads, or ESPN.” *Id.*, 8:18-21.

Therefore, Avellan discloses that the requested media stream “includes a plurality of multimedia items of different types.” With respect to this element, the Patent-at-Issue states that “[t]he media stream can include any number of multimedia items such as audio/video multimedia items” and that “the request can include a channel identifier that is associated with a particular category of multimedia items, such as a music genre corresponding to rock, classical, jazz, etc.” Ex-1001, 7:51-56. Similarly, Avellan discloses that the broadcasted media stream can include a plurality of multimedia items of different types, including different genres and categories of live television, movies, sports, music, etc.

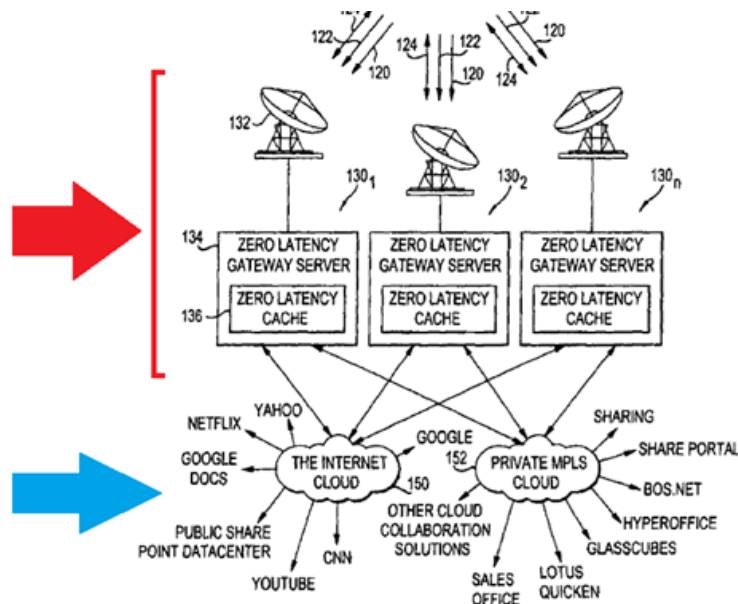
Element 1b

[1b]	obtaining content corresponding to the plurality of multimedia items from at least one source offering the content in at least one first format;
-------------	--

Avellan teaches this element.

Avellan’s “gateway server 134 may be configured to communicate with the public computers, servers, networks or clouds such as the Internet cloud 150,

and/or one or more private computers, servers, networks or clouds 152 such as a private MPLS (MultiProtocol Label Switching) cloud.” Ex-1003, 4:14-18. “In the broadcast mode, the gateway server 134 retrieves the desired data from the networks 150, 152, captures that data (including audio) to a video (or image) frame format, compresses that video data and stores the compressed video data to the gateway 130 at the cache 136.” Ex-1003, 8:42-46. For instance, “[t]he gateway server 134 can track the CNN site on the Internet cloud 150 and store the site to the gateway cache 136 whenever the site is modified and/or at predetermined periodic intervals.” Ex-1003, 10:46-49. Figure 1 shows that networks 150 and 152 provide a range of content, including content from multimedia websites like CNN, Netflix and YouTube (blue).



Ex-1003, Fig. 1. Therefore, Avellan teaches that content corresponding to the plurality of multimedia items is obtained from at least one source, including networks 150 and 152 and/or cache 136, and that content will be in a first format.

Element 1c

[1c]	rendering a web page by a browser using the content;
-------------	--

Avellan teaches this element.

In Avellan “the browser software 106 resides at the gateway 134, and is configured to run as a virtual browser as a remote client of the gateway server 134.” Ex-1003, 4:56-58. Because the browser software resides at the gateway, “the remote browser 106 may rely upon the computing power and very high-speed connection of the gateway server 134 to resolve and render pages, images, and documents from the networks 150, 152.” Ex-1003, 4:62-65.

Element 1d

[1d]	generating a temporal sequence of screen captures of the rendered web page, where each screen capture defines all the content of the web page at a given time, and at least two adjacent screen captures illustrate a dynamic change of at least a portion of the content over time;
-------------	--

Avellan teaches this element.

First, as discussed above, the Avellan gateway 134 includes browser software 106 that renders web pages using the content obtained from the Internet 150 and/or a private network 152 cloud. “The gateway server 134 then images

those web pages (which are typically in an HTTP format) to a video or image frame format.” Ex-1003, 4:65-67. This screen capture process “is preferably done by taking an image of the entire web page from top to bottom, including any portion that would normally be accessed by scrolling up/down or left/right but not immediately displayed on the screen because it is too long for a single screen.” Ex-1003, 4:67-5:4. Therefore, Avellan teaches generating screen captures of the rendered web page, where each screen capture defines all content of the web page at a given time.

Second, Avellan teaches generating a temporal sequence of such screen captures and that at least two adjacent screen captures illustrate a dynamic change of at least a portion of the content over time. Avellan’s “web pages often have information which is rendered over a period of time, such as a video, streaming information, or an animated display. Accordingly, the gateway server 134 continuously captures images of the web page until it detects a repeating pattern of the information in the web page.” Ex-1003, 5:5-10. “Thus, gateway server 134 will be certain to capture all the information on the web page even if that information continues over a period of time.” Ex-1003, 5:10-12.

Avellan discloses using “differential burst video transmission” to more efficiently handle two adjacent screen captures illustrating a dynamic change. Ex-1003, 8:3-11 (“In addition, the differential aspect of the transmission protocol

provides that only the change in the video is transmitted. For instance, suppose the first frame of a video shows a person sitting in a chair. An initial burst transmission of data is transmitted showing the person sitting in the chair. If the person raises his/her leg in the second frame, then the system only needs to transmit the leg being raised and need not transmit the rest of the person's body or the chair. The raising of the leg is sent in a short burst transmission.”).

By continuously capturing screen images of information that is rendered over time—such as “video, streaming information, or an animated display”—two adjacent screen captures will illustrate a dynamic change of at least a portion of the content over time. Ex-1007, ¶122. For example, if a streaming hockey game is being rendered, a screen capture at Time 1 followed by a screen capture at Time 2 will illustrate a dynamic change in that game, such as the movement of players over time—*i.e.*, a dynamic change of at least a portion of the content over time. *Id.*

Element 1e

[1e]	assembling the at least one media stream using the temporal sequence of screen captures; and
-------------	--

Avellan teaches this element.

After capturing screen images of dynamic content as discussed above, Avellan can then “compress the data into a compressed digital video format that can be sent to the user computer 104 for display in the browser 106.” Ex-1003,

5:20-22. Specifically, “[o]nce the page is imaged (including any audio on the web page), the gateway server 134 compresses the image or video frame format and sends it to the browser 106 via the satellite 102.” Ex-1003, 5:12-5:15.

Element 1f

[1f]	providing the at least one media stream to the content provider for broadcast on the broadcast media channel;
-------------	---

Avellan teaches this element.

As discussed above, “[o]nce the page is imaged (including any audio on the web page), the gateway server 134 compresses the image or video frame format and sends it to the browser 106 via the satellite 102.” Ex-1003, 5:12-15. Avellan specifically notes that “suitable sites for the broadcast mode are news sites like CNN, BBC, newspaper, live streaming television feeds, movies, music downloads, or ESPN.” *Id.*, 8:18-21. Therefore, the media stream is provided to the content provider (gateway platform 130) in Avellan for broadcast on the broadcast media channel.

Element 1g

[1g]	wherein the at least one media stream corresponds to at least one of an HTTP live stream (HLS), an HTTP playlist, and a Real-time Streaming Transport (RTSP) stream; and
-------------	--

This element is obvious in view of Avellan. Ex-1007, ¶128.

First, as discussed in conjunction with [1a] and [1b], Avellan discloses obtaining content corresponding to the plurality of multimedia items included in a media stream.

Second, Avellan discloses that “a web-to-HD video protocol may be used to capture and transmit images/audio of a web page, including a full multimedia web page, regardless of its size or original format (e.g., HTTP).” Ex-1003, 5:16-19. Specifically, in Avellan “[t]he gateway cache 136 preferably stores the web pages in the compressed video format, but can store the data in any suitable format(s) including HTTP format.” Ex-1003, 8:50-53. Additionally, Avellan discloses transmission of live content, including “live streaming television feeds” and “live broadcasting.” Ex-1003, 8:18-21; 12:37.

A POSITA would recognize that Avellan contemplates the use of a video transfer protocol and would be motivated to apply a commonly-known transmission protocol³ that was tailored for the delivery of live content. Ex-1007, ¶125. HLS sends audio and video as a series of small files (called media segment files) and an HTTP playlist provides an ordered list of the URLs of the media segment files. Ex-1007, ¶126. The use of HLS to deliver live and on-demand audio and video streams from the internet has been well known since at least 2009. *Id.*

³ During prosecution the Examiner found that this element was not new in view of Luthra [US 2014/0068690] (Ex-1002 at 39 citing Ex-1011 Luthra at [0013]) and applicant did not traverse. Ex-1007, ¶127.

Similarly, RTSP is a network protocol that controls the streaming of media content over the internet and has been well known since at least 1998. *Id.* HLS and RTSP would have been obvious design choices as standard protocols that enable Quality of Service (QOS) management of variable bit rate video over the internet and a POSITA would have used them with a reasonable expectation of success. *Id.* Accordingly, in light of Avellan alone a POSITA would understand that one way to deliver Avellan’s “live streaming television feeds” would be to use well-known and widely used tools such as HLS or RTSP. *Id.*

Element 1h

[1h]	wherein the web page is rendered in parallel in multiple threads.
-------------	---

This element is obvious in view of Avellan.

First, as shown in [1c], Avellan discloses rendering a web page by a browser using the content. Ex-1007, ¶129.

Second, Avellan further states that the disclosed steps of Fig. 2 “may be performed in parallel.” Ex-1003, 14: 4-7. Avellan also references Java and acknowledges that it was well known in the art. Ex-1003, 13:40-41. Java is a programming language that can process multiple threads in parallel and Java would have been well known to POSITA. Ex-1007, ¶129. Given that Avellan discloses performing the process in parallel, a POSITA would have been motivated to use Java to execute multiple threads in parallel in Avellan. *Id.* It would have been

beneficial to use multiple threads that operate in parallel in view of Avellan because (1) it would allow faster rendering by a browser; and (2) it would facilitate efficient use of multi-processor cores that were ubiquitous as of the priority date of the Patent-at-Issue. *Id.*

b. Claim 2

Element 2

[2]	The method according to claim 1, wherein the at least one media stream is interacted with MPEG two way control messages.
-----	--

This element is obvious in view of Avellan. Ex-1007, ¶132.

First, Avellan discloses the use of a tag file which “retains the functionality of the website on the browser 106” allowing user interaction. Ex-1003, 13:45-46. The tag file maps the desired website and includes positions of the scroll bar click, prompt locations, and other action points on the website. Ex-1003, 13:43-59. Therefore, Avellan implicitly discloses the use of two way control messages because these cite items are meant for user interaction. Ex-1007, ¶130.

Second, Avellan discloses that the web page data may be compressed “into a compressed digital video format that can be sent to the user computer 104 for display in the browser.” Ex-1003, 5:19-22. MPEG and its features were well known to POSITA since at least 1995. Ex-1007, ¶90. A POSITA would understand that MPEG is a compressed digital video format that was well known at the time.

Ex-1007, ¶131. That MPEG includes two way control messages was also well known to POSITA. *Id.*

Because Avellan discloses two way interaction and use of compressed digital video formats such as MPEG, it would have been an obvious design choice for a POSITA to use MPEG two way control messages as a mechanism to facilitate such interaction in Avellan. Ex-1007, ¶132. Accordingly, POSITA would have found it obvious to use MPEG two way control message to interact with the media stream of Avellan. *Id.*

c. Claim 3

Element 3

[3]	The method according to claim 1, wherein MPEG messages are sent to a service provider via a unicast address.
------------	--

This element is obvious in view of Avellan. Ex-1007, ¶133.

Avellan discloses a “unicast mode” where “the user requests unicast data (e.g., less-popular websites or websites having highly user-customized information) at the browser” and the “gateway server images that one web page, compresses the unicast data, and transmits the compressed unicast data over the broadcast device substantially in real time to the one requesting user computer.” Ex-1003, [Abstract]. Avellan further discloses transmitting “digital video images of a web page” to a user in the unicast mode. Ex-1003, 11:3-10. As discussed in

[2], it was well known to a POSITA that MPEG is a standard for compression and transmission of digital video images that has been around since at least 1995. Ex-1007, ¶133. Moreover, it was well known to a POSITA that an MPEG stream includes MPEG messages. *Id.* Therefore, it would have been obvious to a POSITA to send MPEG messages to a service provider via Avellan's unicast mode. *Id.*

d. Claim 4

Element 4 pre

[4 pre]	A computer-implemented method, comprising:
----------------	--

To the extent the preamble is limiting, Avellan teaches it as shown in [1 pre].

Element 4a

[4a]	receiving, from a content provider, a request for at least one media stream for playback on a broadcast media channel, wherein the at least one media stream includes a plurality of multimedia items of different types;
-------------	---

Avellan teaches this element as shown in [1a].

Element 4b

[4b]	obtaining content corresponding to the plurality of multimedia items from at least one source offering the content in at least one first format;
-------------	--

Avellan teaches this element as shown in [1b].

Element 4c

[4c]	rendering a web page by a browser using the content;
-------------	--

Avellan teaches this element as shown in [1c].

Element 4d

[4d]	generating a temporal sequence of screen captures of the rendered web page, where each screen capture defines all the content of the web page at a given time, and at least two adjacent screen captures illustrate a dynamic change of at least a portion of the content over time;
-------------	--

Avellan teaches this element as shown in [1d].

Element 4e

[4e]	assembling the at least one media stream using the temporal sequence of screen captures; and
-------------	--

Avellan teaches this element as shown in [1e].

Element 4f

[4f]	providing the at least one media stream to the content provider for broadcast on the broadcast media channel;
-------------	---

Avellan teaches this element as shown in [1f].

Element 4g

[4g]	wherein assembling the at least one media stream comprises inserting at least one MPEG packet including metadata corresponding to the multimedia item; and
-------------	--

This element is obvious in view of Avellan. Ex-1007, ¶134.

Avellan discloses that the web page data may be compressed “into a compressed digital video format that can be sent to the user computer 104 for display in the browser.” Ex-1003 at 5:19-22. A POSITA would understand that MPEG is a compressed digital video format that was well known at the time. Ex-1007, ¶134. Moreover, it was well known to POSITA that MPEG packets include metadata corresponding to the multimedia item. *Id.* Therefore, a POSITA would have found this limitation obvious in view of Avellan. *Id.*

Element 4h

[4h]	wherein the web page is rendered in parallel in multiple threads.
-------------	---

This element is obvious in view of Avellan as shown in [1h].

e. Claim 5

Element 5

[5]	The computer-implemented method of claim 4, wherein the at least one media stream is interacted with MPEG two way control messages.
------------	---

This element is obvious as shown in [2].

f. Claim 6

Element 6

[6]	The computer-implemented method of claim 4, wherein MPEG messages are sent to a service provider via a unicast address.
------------	---

This element is obvious as shown in [3].

g. Claim 13**Element 13 pre**

[13 pre]	A computer-implemented method, comprising:
-----------------	--

To the extent the preamble is limiting, Avellan teaches it as shown in [1 pre].

Element 13a

[13a]	receiving, from a content provider, a request for at least one media stream for playback on a broadcast media channel, wherein the at least one media stream includes a plurality of multimedia items of different types;
--------------	---

Avellan teaches this element as shown in [1a].

Element 13b

[13b]	obtaining content corresponding to the plurality of multimedia items from at least one source offering the content in at least one first format;
--------------	--

Avellan teaches this element as shown in [1b].

Element 13c

[13c]	rendering a web page by a browser using the content;
--------------	--

Avellan teaches this element as shown in [1c].

Element 13d

[13d]	generating a temporal sequence of screen captures of the rendered web page, where each screen capture defines all the content of the web page at a given time, and at least two adjacent screen captures
--------------	--

	illustrate a dynamic change of at least a portion of the content over time;
--	---

Avellan teaches this element as shown in [1d].

Element 13e

[13e]	assembling the at least one media stream using the temporal sequence of screen captures; and
--------------	--

Avellan teaches this element as shown in [1e].

Element 13f

[13f]	providing the at least one media stream to the content provider for broadcast on the broadcast media channel;
--------------	---

Avellan teaches this element as shown in [1f].

Element 13g

[13g]	wherein the at least one media stream corresponds to at least one of an HTTP live stream (HLS), an HTTP playlist, and a Real-time Streaming Transport (RTSP) stream; and
--------------	--

This element is obvious in view of Avellan as shown in [1g].

Element 13h

[13h]	wherein the web page is rendered in parallel in virtual instances or virtual machines.
--------------	--

The web page being rendered in parallel is obvious in view of Avellan as shown in [1h].

Additionally, Avellan discloses using a virtual browser and therefore, teaches that the web page is rendered in virtual instances or virtual machines. Ex-1003 at 3:27-34 (“The user utilizes a virtual browser 106 at a display of one of the user computers 104₁-104_n, to access the remote networks 150, 152 through the broadcast device 102 and the remotely-located gateway platform 130₁-130_n. The virtual browser 106 controls the gateway server 134 and is a remote client of the remotely-located gateway server 134. Thus, the user computers 104₁-104_n can be a dumb terminal which runs the browser 106.”); 4:18-25 (“Generally, the system 100 provides for the user computers 104-104, to access the Internet 150 and/or private network cloud 152 through a virtual browser 106 of the gateway 130. The user operates the browser 106 which sends a signal over the satellite 102 to the gateway 130. The gateway 130 accesses the networks 150, 152 and returns the results to the virtual browser 106 via the Satellite 102.”); 4:56-61 (“As mentioned above, the browser software 106 resides at the gateway 134, and is configured to run as a virtual browser as a remote client of the gateway server 134. Because the browser 106 functions as a virtual browser, viruses that attack operating systems and threaten users’ security are unable to infest the users individual computing devices.”); 6:55-58 (“It will be apparent that the virtual browser 106 can be any software application which can display the received video frame data to look like the web page is actually being displayed in a browser.”).

Moreover, the use of a “virtual machine” or a “virtual instance” to render a webpage was well known to POSITA and would have been an obvious design choice. Ex-1007, ¶137 (*see, e.g.*, Ex-1005 at 7, 104 (disclosing the use of a Java Virtual Machine (“JVM”) and Adobe Flash Player). A POSITA would recognize that it would be beneficial to use such a virtual machine because doing so would reduce system requirements at the user device. *Id.*

B. Ground 2: Claims 1-6, 13 are obvious in view of Avellan, Ma, and Fain.

a. [1pre]-[1f]

As discussed above in conjunction with Ground 1, Avellan discloses these limitations.

b. [1g]

This element is obvious in view of Avellan with Ma. Ex-1007, ¶144.

Ma discloses a “live stream processor 102 [that] is responsible for acquiring source content from a live feed and preparing the content for distribution.” Ex-1004, [0033]. The “source content” of Ma includes media streams such as “real-time audio/video content.” Ex-1004, [0031]. Ma further discloses that:

In one embodiment, the prepared output files are segmented into fixed duration segment files (e.g., MPEG-TS segments, fragmented MP4 segments, 3GP DASH segments, etc.) . . . In one embodiment, the live stream processor 102 generates manifest files indicating

segment retrieval locations for each encoding (e.g., HLS m3u8, DASH MPD, etc.).

Ex-1004, [0033]. M3u8 is an HTTP playlist file used to define a multimedia playlist for HLS. Ex-1007, ¶140. Ma further discloses a “live stream recorder 202” that is “responsible for attaching to the live feed using various protocols” including HLS. Ex-1004, [0045]. Therefore, Ma discloses having a media stream that corresponds to HLS and an HTTP playlist. *Id.*

A POSITA would have been motivated to combine Avellan and Ma and use a media stream that corresponds to one of HLS, HTTP playlist and RTSP. Ex-1007, ¶141. A POSITA would have been motivated to do so because the HLS protocol is a well-known protocol that provides advantages for streaming live content, including supporting adaptive bit rate streaming which can adjust the video quality based on bandwidth and device capabilities without interrupting the stream. *Id.* Moreover, HLS is supported by most Internet devices that transmit video via HTTP, which makes it simple to implement and interoperable. *Id.*

Avellan teaches that “suitable sites for the broadcast mode are news sites like CNN, BBC, newspaper, live streaming television feeds, movies, music downloads, or ESPN.” Ex-1003, 8:18-21. Moreover, as disclosed in Avellan, the gateway cache 136 can store the data in any suitable format(s) including HTTP format. Ex-1003, 8:50-53. Additionally, Avellan discloses transmission of live content, including “live streaming television feeds” and “live broadcasting.” Ex-

1003, 8:18-21; 12:37. As discussed above, HLS and RTSP are standard protocols well known to POSITA and widely used long before the filing date of the Patent-at-Issue. Ex-1007, ¶142. Ma is directed to performing “targeted advertisement replacement for near-real-time streaming media.” Ex-1004, [0001]. Both Ma and Avellan are generally directed to providing web content in a media stream to subscribers. *Id.* A POSITA would recognize that it would be beneficial to use a media stream corresponding to a standard protocol such as HLS or RTSP as disclosed by Ma in Avellan since the use of such standardized protocols allows interoperability and simplifies implementation. *Id.* Therefore, a POSITA would have been motivated to combine the teachings of Ma with Avellan. *Id.* A POSITA would have had a reasonable expectation of success because HLS can be implemented on a server that uses HTTP such as in Avellan as a standard transport protocols that would simply the transmission of Avellan’s media streams. Ex-1007, ¶143.

Accordingly, in light of a combination of Avellan with Ma, a POSITA would understand that one way to deliver Avellan’s “live streaming television feeds” would be to use well-known and widely used protocols such as HLS or RTSP. Ex-1007, ¶144.

c. [1h]

This element is obvious in view of combination of Avellan, Fain, and Ma.

Fain is a book titled “Java Programming 24-Hour Trainer” and includes a chapter titled “Introduction to Multi-Threading” which discusses Java’s multi-threading capability stating, for example, that “a Java program can start multiple threads of execution that will run in parallel.” Ex-1005 at 209. Fain adds that “[w]eb browsers maintain two or more connections to any website, and can download multiple resources (text, images, videos, music) at the same time. Any web browser works in a multi-threaded mode.” Ex-1005, 209. Therefore, it would have been obvious to combine Avellan’s capture and distribution with Fain’s multi-threading distribution in parallel to more efficiently deliver the types of digital multimedia content disclosed in Avellan. Ex-1007, ¶145.

A POSITA would have been motivated to combine Avellan and Fain. Ex-1007, ¶146. Avellan discloses rendering web content and provides that the disclosed steps “may be performed in parallel.” *Id.*; Ex-1003, 14: 4-7. Avellan also references Java and acknowledges that it was well known. *Id.*; Ex-1003, 13:40-41. Java was a well-known programming language to POSITA that allowed for parallel processing of multiple threads. *Id.* A POSITA would have been motivated to combine the two to implement the multiple parallel Java threads in Avellan because doing so would be inherently efficient in a multiprocessor system where “parallel threads can run on different CPUs” and “[e]ven if you have only one processor ... it could still be a good idea to run some tasks in parallel.” *Id.*; Ex-

1005, 209-210. A POSITA would have had a reasonable expectation of success in combining Avellan with Fain's parallel threads to facilitate efficient content delivery. *Id.*

The use of Ma with respect to the separately recited function of [1g] is not affected by Fain and would not affect Fain, as they are directed towards different aspects of the claim. Therefore, no additional motivation to combine Avellan and Fain with Ma is required. Nevertheless, it would have been obvious to use both the standard protocol of Ma as discussed in [1g] and the parallel multiple threads of Fain in conjunction with Avellan. Ex-1007, ¶147.

Therefore, it would have been obvious to a POSITA to render the Avellan web page in parallel in multiple threads considering Avellan in view Fain and Ma. Ex-1007, ¶148.

d. [2]

This element is obvious in view of Avellan, Fain and Ma.⁴ Ex-1007, ¶153.

The only “two way” control message referenced in the specification is the MPEG TS Control messages. Ex-1001, 11:35-40 (“For example, the metadata added could be as simple as a string containing Artist, Name, or other information, or as complex as DCII Text (DigiCipher version 2, as specified by Motorola),

⁴ Fain as used in 1[h] is directed to a non-overlapping claim element and does not affect the combination of Avellan and Ma. Moreover, as discussed in [1h], a POSITA would have been motivated to combine Avellan with Fain and Ma.

MPEG TS Control messages (Two-way), or any other metadata or PID that can be inserted into the stream being sent to the multicast address and port.”).

Ma discloses the use of MPEG-TS (i.e., MPEG two way control messages) for interaction with the media stream. Ex-1007, ¶150; Ex-1004 at [0050] (“The inline metadata insertion module 210 is responsible for inserting upcoming advertisement break notifications and exact advertisement start and end notification into the segments. ... There are multiple segment file formats (e.g., *MPEG-TS*, DASH, RTP, etc.)”); [0058] (“There are multiple segment file formats (e.g., *MPEG-TS*, DASH, RTP, etc.)”). Ma further discloses inserting metadata (e.g., ID3 tags) into MPEG-TS segments and using this data for providing advertisements. Ex-1004 at [0038]-[0041]. For instance, Ma discloses that “the client 110 detects advertisement metadata by monitoring callbacks from the media player (e.g., receiving ID3 callbacks).” Ex-1004 at [0041]. Ma further discloses using metadata (e.g., ID3 callbacks) and DASH manifests to request content by the media player. Ex-1004 at [0038]-[0042]. Therefore, Ma discloses MPEG metadata that interacts with two way broadcast TV systems—i.e.—that the at least one media stream is interacted with MPEG two way control messages. *Id.*

Indeed, the Examiner rejected this element in view of Ma during prosecution and the Applicant did not traverse that rejection. Ex-1002, 40.

A POSITA would have been motivated to combine Avellan with Ma and would have had a reasonable expectation of success doing so. Ex-1007, ¶152. Avellan is directed to a communication system that can “broadcast information from a web server over a broadcast device to a multitude of user computers.” Ex-1003, 2:9-10. Ma is directed to performing “targeted advertisement replacement for near-real-time streaming media.” Ex-1004, [0001]. Hence, both Ma and Avellan are generally directed to providing web content in a media stream to subscribers. Ex-1007, ¶152. Further, Avellan discloses that the web page data may be compressed “into a compressed digital video format that can be sent to the user computer 104 for display in the browser.” Ex-1003, 5:19-22. A POSITA would understand that MPEG is a compressed digital video format that was well-known at the time and Ma explicitly discloses the use of MPEG. Ex-1007, ¶152. Therefore, a POSITA would have been motivated to combine the teachings of Ma with Avellan and modify Avellan to utilize Ma’s control messages in order to provide customized targeted content to a user which would provide content of most interest to that user. *Id.*; Ex-1002 at 40.

e. [3]

This element is obvious in view of Avellan, Fain and Ma.⁵ Ex-1007, ¶156.

⁵ Fain as used in 1[h] is directed to a non-overlapping claim element and does not affect the combination of Avellan and Ma. Moreover, as discussed in [1h], a POSITA would have been motivated to combine Avellan with Fain and Ma.

Ma discloses using MPEG messages in a unicast mode. Specifically, Ma discloses “live stream processor 102 includes multiple components, including a live stream recorder 202 which records source audio/video content.” Ex-1004, [0044]. Ma adds that “[t]he live stream recorder 202 is responsible for attaching to the live feed using various protocols (e.g., *unicast MPEG-TS*, multi-cast MPEG-TS, unicast RTP, multi-cast RTP, HLS, DASH, RTMP, etc.).” Ex-1004, [0045].

For the same reasons discussed in [2], a POSITA would have been motivated to combine Avellan and Ma to implement the unicast MPEG messages of Ma in the unicast mode of Avellan because it would allow user control of media stream once playback starts. Ex-1007, ¶155.

f. [4pre]-4[f]

As discussed above in conjunction with Ground 1, Avellan discloses these limitations or renders them obvious.

g. [4g]

This element is obvious in view of Avellan, Fain and Ma.⁶ Ex-1007, ¶160.

First, Ma discloses this limitation. Ex-1007, ¶157. Specifically, Ma discloses a metadata insertion module that supports different file formats including MPEG-TS. *See* Ex-1004, [0050] (“The inline metadata insertion module 210 is

⁶ Fain as used in 1[h] is directed to a non-overlapping claim element and does not affect the combination of Avellan and Ma. Moreover, as discussed in [1h], a POSITA would have been motivated to combine Avellan with Fain and Ma.

responsible for inserting upcoming advertisement break notifications and exact advertisement start and end notification into the segments. ... There are multiple segment file formats (e.g., MPEG-TS, DASH, RTP, etc.)”). Ma further discloses inserting metadata (e.g., ID3 tags) into MPEG-TS segments and using this data for providing advertisements. *Id.*, [0038]-[0041]. Ma further discloses inserting metadata (e.g., ID3 tags) into MPEG-TS segments and using this data for providing advertisements. *Id.*, [0038]-[0041].

Second, the Examiner rejected this element in view of Ma during prosecution and Applicant did not traverse that rejection which further confirms that this element would have been obvious in view of Ma. Ex-1002, 40-41.

For the same reasons discussed in [2], a POSITA would have been motivated to combine Avellan and Ma to modify Avellan “to include the MPEG and metadata features of Ma for the advantages of providing targeted advertising replacement (Ma at [0041]) and utilizing a well-known standard that is interoperable with many devices.” Ex-1002, 41; Ex-1007, ¶159. For instance, doing so would facilitate better conversion rate and more efficient add placement. *Id.*

h. [4h], [5], [6]

These elements would have been obvious for the same reasons provided with respect to [1h], [2], and [3], respectively.

i. [13pre]-[13f]

As discussed above in conjunction with Ground 1, Avellan discloses these limitations or renders them obvious.

j. [13g]

This element is obvious in view of Avellan with Ma as shown in [1g].

k. [13h]

As discussed in [1h], rendering a webpage in parallel would have been obvious in view of combination of Avellan, Fain, and Ma. As discussed in Ground 1, 13[h], Avellan discloses that webpage is rendered in virtual instances or virtual machines. Moreover, Fain also discloses this limitation. Ex-1005, 7, 104 (disclosing the use of a Java Virtual Machine (“JVM”) and Adobe Flash Player).

C. Ground 3: Claims 7-9 and 11-12 Are Anticipated by Wannamaker.

a. Claim 7

Element 7 pre

[7 pre]	A system, comprising:
----------------	-----------------------

To the extent the preamble is limiting, Wannamaker discloses this element.

Wannamaker discloses a “client/server system [that] provides a platform for cable system operators to deliver a full range of sophisticated new products to cable television subscribers.” Ex-1006, [Abstract].

Element 7a

[7a]	a multicast server configured to generate parameters relating to at least one of video content, image content and audio content, and configured to output the parameters via MPEG outputs;
-------------	--

Wannamaker discloses this element.

Wannamaker's application server includes "the data carousel server for data formatting, and Multimedia Streamer (MMS) components for data transmission." Ex-1006, [0169]. "The data carousel server creates objects in the Mediacast signaling protocol (MSP) format, and uses the multimedia streamer (MMS) components to transport the MSP objects to the downstream network." Ex-1006, [0174].

First, the application server 24 of Wannamaker which includes the data carousel server is a multicast server and includes the MMS components for data transmission. Ex-1006, [0169, 175, 181]. The out-of-band servers of the MMS can "multicast to a group of set-top boxes." Ex-1006, [0229-0232]. Moreover, with respect to the operation of the microbrowser, Wannamaker discloses that "transcoded content is ... inserted into [a] multicast file system." Ex-1006, [0490].

Second, the application server 24 of Wannamaker is configured to generate parameters relating to at least one of video content, image content and audio content at the cache manager, at the data carousel as well as in conjunction with

background images, a microbrowser, and size of assets when operating in the “layout preview mode.”

The cache manager sends a message to all cache managers on content processors hosts to resolve a URL. Ex-1006, [0236] (“When a process needs to fetch content, it requests the local cache manager to resolve the URL for the content.”); [0237] (If the content is not local, “the cache manager sends a message to all cache managers on content processor hosts that have been configured to resolve URL requests.”). The URL to be resolved may correspond to images, animated GIFs or other elements. Ex-1006, [0275] (“... Within the directory is the CPML file, the main FMB file containing rendered images, and one or more subdirectories containing the rendered assets (animated GIFs, form elements) for the given CPML file.”). Therefore, this message from the cache manager on the application server 24 to the cache manager on the content processors 23 is a parameter relating to video content, image content, and/or audio content corresponding to the particular URL.

Additionally, the “[t]he data carousel server creates objects in the Mediacast signaling protocol (MSP) format.” Ex-1006, [0174]. The MPS objects are of different types and include “Directory Objects” which “form a list of resource locations, and can reference other directory objects, data objects, audio and video

resources.” Ex-1006, [0187]. Accordingly, the MPS objects are parameters relating to at least one of video content, image content and audio content.

The application server also stores background graphics which are converted from JPEG into an MPEG I-frame allowing the set-top box to render them. Ex-1006, [0567] (“The invention also enables content developers to include aesthetically pleasing JPEG images into the background of an application. The I-frame can be a standard 24-bit JPEG file, which is stored on the application server and sent to the transcoder. The transcoder converts this JPEG file into an MPEG I-frame, allowing the set-top box to render it.”).

Wannamaker also discloses a “microbrowser for broadcast managed content.” Ex-1006, [0009]. “The microbrowser displays and navigates HTML content that has been transcoded by the content processor and then broadcast via the Mediacast component” which as noted above (and shown in Fig. 3), is part of the application server 24. *Id.*, [0112]. For microbrowser rendering, the “information platform uses the thin client model to provide a Web browser on digital set-top boxes.” *Id.*, [0268]. The “server fetches content from the Web and performs the initial translation, layout, and rendering. The server caches the converted content for future reuse, and crawls the Web as configured by the server administrator to refresh the cached data.” *Id.*, [0270]. “The server continuously

transmits the Web data to the client in an MPEG video stream [and] puts images of the rendered Web page in MPEG I-frames and MPEG P-frames.” *Id.*, [0271].

Finally, Wannamaker discloses that the Mediacast server broadcasts “the size (in bytes) of each asset” produced by a page. Ex-1006, [0261] (In lay out preview mode “there is additional information displayed below the screen capture [including] the size (in bytes) of each asset, produced by this page, that are broadcast by the Mediacast server”). Accordingly, the size of each asset is another parameter that is generated relating to at least one of video content, image content and audio content on a page. Ex-1007, ¶169.

Third, the application server of Wannamaker is configured to output the parameters via MPEG outputs. Specifically, the data carousel server includes the MMS components for data transmission. Ex-1006, [0169]. The “multimedia streamer (MMS) consists of the audio/video/terminal server (AVTS) that delivers the units of audiovisual information sent by the system to the user’s set-top box.” Ex-1006, [0193]. “The AVTS is responsible for delivering MPEG video and Dolby AC-3 audio streams to the set-top box. A special MPEG multiplexer (MUX) card, controlled by the AVTS, streams data onto the cable network.” Ex-1006, [0195].

Element 7b

[7b]	a caching server communicatively coupled to the multicast server,
-------------	---

Wannamaker discloses this element.

Wannamaker's application server 10 "comprises an application server processor 24 and a content processor 23" and includes "caching and transaction services." Ex-1006, [0054], [0059]. The content processor "produces digital cable system compatible (rendered) content from standard Web content retrieved from the Internet." *Id.*, [0244]. "CP processes involved in accessing the Internet and rendering content are the scheduler, crawler, and renderer server and renderer agents." *Id.*, [0244]. "The scheduler ensures that content is fetched, rendered, and copied to the appropriate locations for use [and] triggers an update through the cache manager. [which], in turn, manages the fetching (via the crawler) and the rendering (via the renderer) on a scheduled basis." *Id.*, [0256]. "The crawler places "the crawled assets in the appropriate source cache directory on the server." *Id.*, [0249]. "The renderer converts the content into a format compatible with the user's set-top box hardware...." *Id.*, [0259].

The content processor 23 which is the "caching server" is communicatively coupled to the application servers 24 (i.e., multicast server) which also has a cache manager 40. *See* Ex-1006, [0099] ("Typical data flows are indicated by the arrows connecting the components").

Element 7c

[7c]	and configured to receive and store the parameters,
-------------	---

Wannamaker discloses this element. Ex-1007, ¶171-175.

Wannamaker's content processor 23 is configured to receive and store parameters generated by the application server 24 as shown, for example, by the arrow between cache manager 40 and cache manager 37 in Figure 3. Wannamaker discloses receiving and storing the parameters on the cache server. Ex-1006, [0098]-[0099].

“Cache managers 37 (see FIG. 3) are responsible for accepting requests for content and handling the steps required to deliver the content to the necessary location.” Ex-1006, [0236]. As discussed with respect to [7a], “[t]o resolve the URL, the cache manager sends a message to all cache managers on content processor hosts that have been configured to resolve URL requests.” Ex-1006, [0237]. In response, “[t]he content processor cache managers all check their local caches for the requested URL and reply with the timestamps and status for their local source and rendered caches.” Ex-1006, [0237]. “The cache manager then implements information platform's smart caching by evaluating the status of all the caches and selecting one of the following mechanisms to best fulfill the application's request: If valid content resides in a rendered cache of one of the content processors, the cache manager immediately copies the content to the rendered caches[;] If the valid content is in a source cache, but not rendered, the cache manager first directs the cache manager on the content processor to render it and then copies the content to the rendered caches[; and] If the content is not in a

source cache, the cache manager directs the least busy crawler to get the source and render it, and then copies the content to the rendered caches.” Ex-1006, [0238]-[0241].

Therefore, the cache managers of the content processor (“cache server”) receive the message (“parameter”) from the application server (multicast server). Ex-1007, ¶173. Moreover, it is inherent that the message would be stored in order to allow the content processor to resolve the requested URL because the system would not be operational otherwise. *Id.*

Additionally, Wannamaker discloses that in the “layout preview mode” the renderer of the content processor receives “the size (in bytes) of each asset” produced by a page from the Mediacast server of the application server. Ex-1006, [0261] (“The renderer also supports a layout preview mode that emulates a system running the microbrowser. ... In this mode, there is additional information displayed below the screen capture. It includes the size (in bytes) of each asset, produced by this page, that are broadcast by the Mediacast server ...”). Accordingly, the size of each asset is another parameter that is generated relating to at least one of video content, image content and audio content on a page and received by the caching server from the application server. Ex-1007, ¶174. Moreover, in this “layout preview mode”, this information is “displayed below the

screen capture.” It is therefore stored on the display and it is inherent that “the size (in bytes) of each asset” is at least stored in corresponding video memory. *Id.*

Finally, in one mode, the crawler of the content processor “only fetches the document if it is newer than the one on the application server, based on the modification dates provided by the Web server.” Ex-1006, [0254]. This further shows that the content processor inherently receives parameters generated by the application server (e.g., the modification date of the content) and stores them to make this comparison. Ex-1007, ¶175.

Elements 7d & 7e

[7d]	create a temporal sequence of screen captures of a rendered web page, where each said screen capture defines content of the rendered web page at a given time, and at least two adjacent screen captures in the temporal sequence illustrate a dynamic change of at least a portion of the video content, image content or audio content over time,
[7e]	assemble the temporal sequence of screen captures, and

Wannamaker discloses this element.

The “content processor (CP) cluster produces digital cable system compatible (rendered) content from standard Web content retrieved from the Internet.” Ex-1006, [0244]; [0098] (“The information platform converts Internet content and protocols into forms suitable for TV viewing. The content is fetched from Web sites, converted or rendered by the system, and then cached on [the]

application server.”); [0259] (“The renderer converts the content into a format compatible with the user’s set-top box hardware. It does this by accepting URLs of HTML files that are stored locally, and then parsing and rendering the HTML to produce output image files (FMB) and output CPML or MML.”)

This includes web content including animated GIF and other animated images which would necessitate the use of generation of the claimed “temporal sequence” to illustrate a dynamic change in the content. Ex-1006, [0098] (“The information platform supports content from standard Web sites, e.g. HTML source, GIF, animated GIF, JPEG, and PNG files.”); [0136] (“The information platform supports HTML 3.2, including text, tables, forms, images, animated images, image maps, frames, and background images...”).

As an illustrative example, Wannamaker discloses capturing a temporal sequence of screen captures with respect to an animated GIF in separate frames as claimed. Ex-1006, [0263] (“The information platform layout engine processes HTML documents and allows expansion to other document types. All requests to render HTML pages result in a translation of these pages to CPML, and the production of an FMB file containing the MPEG compressed page information. ... The information platform layout engine placement policies produce TV-centric layout of document elements, optimally arranging the content in the output buffer prior to compression to MPEG format ... All important page assets are described

by using the CPML constructs such as anchors, form elements, and animated GIFs."); [0275] ("Within the directory is the CPML file, the main FMB file containing rendered images, and one or more subdirectories containing the rendered assets (animated GIFs, form elements) for the given CPML file. Taking this example a step farther, if there is an animated GIF on the same page, with a URL of <http://www.liberate.com/images/anim.gif>, it is rendered as separate frames into the /prefix/www.liberate.com/images/anim.gif/directory. Each frame is called fmb.TIMESTAMP.sequence_number. The same scheme holds for all special assets of the presentation (animated GIFs, forms, frames).").

To accomplish the creation and assembly of the claimed temporal sequence of screen captures, Wannamaker discloses use of MPEG I-frames and P-frames. Ex-1006, [0264] ("After layout, the information platform renderer quickly and effectively compresses the output buffer to MPEG syntax in the FMB file format."); [0271] ("The server continuously transmits the Web data to the client in an MPEG video stream. The server puts images of the rendered Web page in MPEG I-frames and MPEG P-frames.).

Element 7f

[7f]	provide at least one of the video content, image content and audio content to the multicast server for generating a multicast stream with MPEG encoded video and audio; and
-------------	---

Wannamaker discloses this element. Ex-1007, ¶176-182.

The application servers 24 of Wannamaker is provided with at least one of video content, image content and audio content for generating a multicast stream with MPEG encoded video and audio. As discussed in further detail below, Wannamaker discloses that “[t]he downstream communication, from server to set-top box delivers an MPEG transport stream containing *MPEG video*, *MPEG audio*, and data via a specialized hardware multiplexer card.” Ex-1006, [0117].

First, as discussed in 7[a], the application servers of Wannamaker generates a multicast stream.

Second, the application server of Wannamaker is provided with at least one of video content, image content, and audio content from the content processor for generating MPEG encoded video and audio at the Mediacast data carousel. Specifically, “the data carousel server creates objects in the Mediacast signaling protocol (MSP) format, and uses the multimedia streamer (MMS) components to transport the MSP objects to the downstream network.” Ex-1006, [0174]. The MPS objects “can reference other directory objects, data objects, audio and video resources.” *Id.*, [0187]. The MMS “consists of the *audio/video*/terminal server (AVTS) that delivers the units of audiovisual information sent by the system to the user’s set-top box.” *Id.*, [0193].

“The AVTS is responsible for delivering MPEG video and Dolby AC-3 audio streams to the set-top box. A special MPEG multiplexer (MUX) card,

controlled by the AVTS, streams data onto the cable network.” Ex-1006, [0195]. The AVTS provides the following: Ability to play full motion video files; Ability to play MPEG I-frames (full-screen images); Ability to play MPEG B-frames (MPEG ‘layer’); Ability to play MPEG replacement P-frames (subscreen sized images); Ability to play MPEG motion vector P-frames.” *Id.*, [0196]-[0201]. The MPEG Card (LMC) “is a PCI-based adapter card that is MPEG-2 transport stream compliant and can concurrently transmit and receive MPEG-2 transport streams.” *Id.*, [0210]. It is well known to a POSITA that MPEG-2 is a standard for coding moving pictures and associated audio information. Ex-1007, ¶179.

Third, this element is met by Wannamaker’s microbrowser feature.

“The microbrowser displays and navigates HTML content that has been transcoded by the content processor and then broadcast via the Mediacast component” which as noted above (and shown in Fig. 3), is part of the application server 24. Ex-1006, [0112]. For microbrowser rendering, the “information platform uses the thin client model to provide a Web browser on digital set-top boxes.” *Id.*, [0268]. “The server continuously transmits the Web data to the client in an MPEG video stream [and] puts images of the rendered Web page in MPEG I-frames and MPEG P-frames.” *Id.*, [0271]. It is well known to a POSITA that MPEG is a standard for coding moving pictures and associated audio information. Ex-1007, ¶180.

Fourth, Wannamaker discloses that the renderer, which is part of the content processor, accepts URLs of HTML files that are stored locally and produces output image files (FMB) and outputs CPML or MML. Ex-1006, [0259] The HTML pages are translated to CPML and result in the production of an FMB file containing the MPEG compressed page information. *Id.*, [0263]. “All important page assets are described by using the CPML constructs such as anchors, form elements, and animated GIFs.” *Id.*, [0263]. “The FMB file information is efficiently transcoded to one or more MPEG frames for transmission to digital set-top boxes.” *Id.*, [0264]. The MMS component of the applications servers transmits the MPEG audio and video to the set top boxes. *Id.*, [0169]; [0193]; Fig. 3.

Further, Wannamaker discloses that “[t]he digital set-top box already contains dedicated MPEG hardware for video and audio decoding, so it uses the same hardware to decode the MPEG data from the server.” Ex-1006, [0272].

Element 7g

[7g]	a monitoring system configured to maintain generation of video and audio files, even in failure cases.
-------------	--

Wannamaker discloses this element. Ex-1007, ¶183-184.

Specifically, Wannamaker discloses an “operator management interface” (OMI) that provides fault tolerance. Ex-1006, [0306]; [0320] (“The system is also capable of detecting fatal faults and restarting itself without human intervention.

Much of the fault tolerance is configurable by the system operator”); [0101] (“Multiple servers can be replicated within the application server, forming one logical server. ... If a server fails, transactions are routed to another server in the server group.”); [0103] (“Fault tolerance is an important aspect of any online high-availability system. The information platform is configured to handle many types of system failures—both software and hardware. ... Note that the application server also includes the ability to load-balance between active servers so that, in addition to failover support, the servers in a cluster can be maximized for increasing overall performance. The network itself may also be configured with redundant network components, such as duplicate switches and interface adapters. As a result of the multiple layers of redundancy, the total system presents very high availability to its users.”); [0579] (“Fault tolerance—three levels of fault tolerance are provided, namely critical hardware, subsystem redundancy, and software fault tolerance.”).

Accordingly, the fault tolerance features of the monitoring system of Wannamaker teach the claimed monitoring system configured to maintain generation of video and audio files, even in failure cases. Ex-1007, ¶184.

b. Claim 8

Element 8

[8]	The system according to claim 7, further comprising multiple
-----	--

	servers for load balancing and fault-tolerance.
--	---

Wannamaker discloses this element.

Wannamaker discloses that “[m]ultiple servers can be replicated within the application server, forming one logical server. ... If a server fails, transactions are routed to another server in the server group. As many servers as necessary can be used to meet the load-balancing requirements of the service.” Ex-1006, [0101]; *see also* [0103] (“Fault tolerance is an important aspect of any online high-availability system. ... Note that the application server also includes the ability to load-balance between active servers so that, in addition to failover support, the servers in a cluster can be maximized for increasing overall performance.”); [0244]; [0320]; [0579].

Wannamaker further discloses that “[s]tandard tools can be used for performance tuning, load balancing, viewing, mirroring, and backing up the database.” Ex-1006, [0325].

c. Claim 9

Element 9

[9]	The system according to claim 7, wherein the multicast server is selected to communicate a multicast stream available for a broadband network.
------------	--

Wannamaker discloses this element.

First, as discussed with respect to [7a], Wannamaker discloses a server that communicates a multicast stream.

Second, as shown in Figure 5, the multicast stream from the application servers is available for a broadband network. Ex-1006, [0122] (“FIG. 5 is a block schematic diagram that shows the information platform, national insertion up link. The application server 10 is installed at the network head-end. Inband content goes through the broadband feed 150 to the encoders 151, then is up linked 152 to the satellite 154.”).

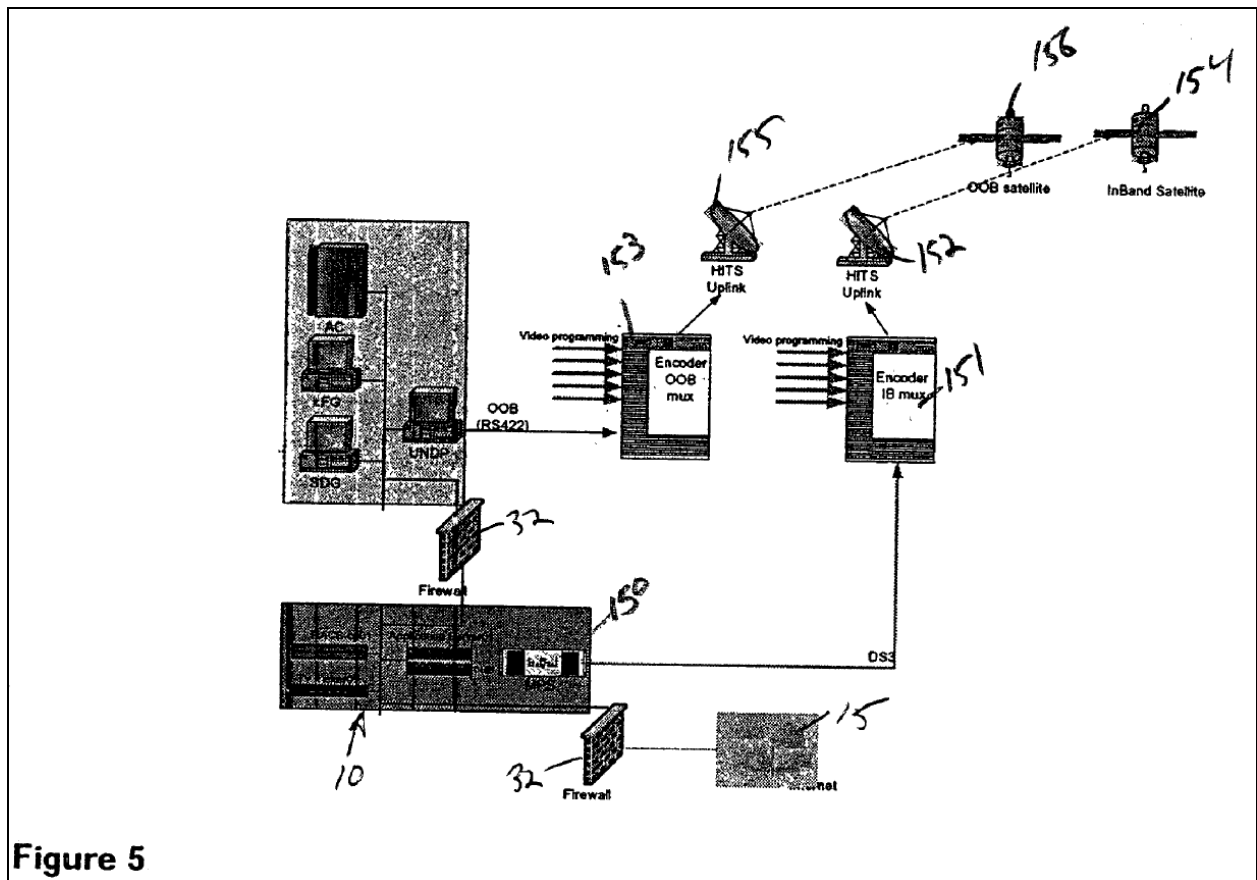


Figure 5

d. Claim 11**Element 11**

[11]	The system according to claim 7, wherein the system generates audio MPEG files and add [sic] metadata in an MPEG frame.
-------------	---

Wannamaker discloses this element. Ex-1007, ¶185-186.

First, Wannamaker discloses that the system generates audio MPEG files. Specifically, Wannamaker discloses that “[t]he downstream communication, from server to set-top box delivers an MPEG transport stream containing *MPEG video*, *MPEG audio*, and data via a specialized hardware multiplexer card.” Ex-1006, [0117]; *see also* [0193]; [0196]-[0201]. The MPEG Card (LMC) “is a PCI-based adapter card that is MPEG-2 transport stream compliant and can concurrently transmit and receive MPEG-2 transport streams.” Ex-1006, [0210]. It is well known to a POSITA that MPEG-2 is a standard that has been around since 1995 and includes a transport stream with video, audio and data streams. Ex-1007, ¶185. The MPEG-2 data stream is designed to carry meta data and program specific information. *Id.* As discussed in further detail with respect to [7f], Wannamaker also discloses the generation of audio MPEG files with respect to the microbrowser.

Second, Wannamaker discloses that the system adds metadata in an MPEG frame. Ex-1006, [0271] (“The server continuously transmits the Web data to the

client in an MPEG video stream. The server puts images of the rendered Web page in MPEG I-frames and MPEG P-frames. In addition, the server transmits some additional information about the Web page's characteristics in an MPEG data stream. This additional information defines foreground text, hotzones, and hyperlink URLs. This is contained in an MML tile that is transcoded from the original HTML.”). Accordingly, a POSITA would understand that Wannamaker discloses adding metadata in an MPEG frame. Ex-1007, ¶ 186.

e. Claim 12

Element 12

[12]	The system according to claim 7, wherein the system generates MPEG metadata that is interacted with two way broadcast TV systems.
-------------	---

Wannamaker discloses this element. Ex-1007, ¶187-189.

First, as discussed with respect to [2], the only “two way” control message referenced in the specification is the MPEG TS Control messages. Ex-1001, 11:35-40. A POSITA would understand that the “MPEG TS Control message” of the Patent-at-Issue refers to the MPEG Transport Stream. Ex-1007, ¶187. Just like the Patent-at-Issue, Wannamaker explicitly discloses the use of the MPEG Transport Stream. Ex-1006, [0117] (“The downstream communication, from server to set-top box delivers an MPEG transport stream containing MPEG video, MPEG audio, and data via a specialized hardware multiplexer card.”); [0191] (“Private-text

packets are then encapsulated into MPEG transport stream packets, and are caroused by the MMS inband or out-of-band MPEG transport streams.”). Transport Streams inherently have Control Messages. Ex-1007, ¶187.

Second, Wannamaker discloses a “microbrowser rendering” feature where the “information platform uses the thin client model to provide a Web browser on digital set-top boxes that would otherwise be too slow.” Ex-1006, [0268]. The system operates as follows:

1. A server fetches content from the Web and performs the initial translation, layout, and rendering. ...
2. The server continuously transmits the Web data to the client in an MPEG video stream. The server puts images of the rendered Web page in MPEG I-frames and MPEG P-frames. In addition, the server transmits some additional information about the Web page’s characteristics in an MPEG data stream. This additional information defines foreground text, hotzones, and hyperlink URLs. This is contained in an MML tile that is transcoded from the original HTML.
3. The digital set-top box already contains dedicated MPEG hardware for video and audio decoding, so it uses the same hardware to decode the MPEG data from the server. ...
4. The client software uses some of these freed-up CPU cycles to run a small but powerful microbrowser that

interprets the contents of the MML tile. MML provides simple commands for drawing lines and text. In combination with the MPEG image data, MML lets the user navigate hypertext links.

Ex-1006, [0269]-[0273].

Wannamaker further discloses that “[n]avigation is restricted to following the hyperlinks” and that a user can use the MPEG metadata to interact with the two way broadcast TV system. Ex-1006, [0278] (“Navigation is restricted to following the hyperlinks. The sites featured within the managed content have disabled the hot links that lead users off the site. Using managed content lets operators and their strategic partners present subscribers with a collection of valuable, aggregated Web sites specifically tailored and enhanced for television viewing. Direct URL entry from a wireless or virtual keyboard is not permitted, and the Internet is only checked for content updates on sites that are registered in the scheduler. This closed content environment offers promotional opportunities for the operator along with a means of generating revenue, as content providers and advertisers pay for exclusive access. All valid managed content is rendered to MPEG and MML format in the content processor cache, as required. The converted content is then posted to the Information Mediacast server for MPEG transmission and display on the user's set-top box.”); [0490] (“The microbrowser 194 (FIG. 9) is an LMAE application that can be used to view broadcast managed content that has been

authored in HTML. ... When activated, the microbrowser reads and displays the home URL. These URLs can either be specified in the microbrowser's configuration file, or in an eTV trigger. The user can then navigate the content in much the same manner as regular interactive HTML content. Selecting some links cause the microbrowser to read and display another page of managed content, while other links start an application such as a game.”). All of this navigation allows the user to “interact[] with [the] two way broadcast TV systems.” Ex-1007, ¶189.

D. Ground 4: Claim 10 Is Obvious in View of Wannamaker and Avellan.

a. Claim 10

Element 10

[10]	The system according to claim 7, wherein the video content is encoded in a video streaming format so as to generate a unicast stream.
-------------	---

This element would have been obvious in view of a combination of Wannamaker and Avellan. Ex-1007, ¶191-194.

As described with respect to elements [7f], Wannamaker discloses the use of MPEG. A POSITA would recognize that MPEG is a video coding standard used for encoding a video streaming format. Ex-1007, ¶191. Also, as described with respect to element 7[a], Wannamaker discloses generating a unicast stream noting

that the out-of-band servers of the MMS can “single-cast or ‘addressed’ to a single set-top box.” Ex-1006, [0229-0232]. Further, as described with respect to [7d]/[7e], Wannamaker also discloses encoding of “animated GIFs” and sounds. *See, e.g., Id.*, [0098] (“The information platform supports content from standard Web sites, e.g. HTML source, GIF, animated GIF, JPEG, and PNG files. The majority of HTML 3.2 tags are supported, e.g. text, tables, forms, images, image maps, background images, and sounds.”). In light of Wannamaker’s disclosure, a POSITA would have found the encoding of video content to generate a unicast stream obvious. Ex-1007, ¶191.

Wannamaker in view of Avellan render this limitation obvious. Ex-1007, ¶192.

First, Avellan discloses a unicast mode. Ex-1003, [Abstract] (“In the unicast mode, the user requests unicast data (e.g., less-popular websites or websites having highly user-customized information) at the browser. The gateway server images that one web page, compresses the unicast data, and transmits the compressed unicast data over the broadcast device substantially in real time to the one requesting user computer.”); 7:53-58 (“Since the unicast mode is for customized information, the unicast data is sent in real time without delay, directly from the network could 150, 152 via the gateway platform 130 and the broadcast device 102

to the user computer 104, where it may be immediately displayed (once decompressed at the computer 104).”).

Second, Avellan discloses encoding video content from a website into a video streaming format in the unicast mode. Ex-1003, 4:56-67 (“... In addition, the remote browser 106 may rely upon the computing power and very high-speed connection of the gateway server 134 to resolve and render pages, images, and documents from the networks 150, 152. The gateway server 134 then images those web pages (which are typically in an HTTP format) to a video or image frame format.”); 5:5-15 (“In addition, web pages often have information which is rendered over a period of time, such as a video, streaming information, or an animated display. ... Once the page is imaged (including any audio on the web page), the gateway server 134 compresses the image or video frame format and sends it to the browser 106 via the satellite 102.”); *see also* 7:58-60.

Avellan is directed to a communication system that can “broadcast information from a web server over a broadcast device to a multitude of user computers.” Ex-1003, 2:9-10. Similarly, Wannamaker discloses a “middleware solution” referred to as an “information platform” which “provides a platform for cable system operators to deliver a full range of sophisticated new products to cable television subscribers.” Ex-1006 [Abstract]. Like Wannamaker, the Avellan system “replicates internet websites” and includes “a gateway platform having a

gateway server and a cache.” Ex-1003, [Abstract]. Moreover, both Avellan and Wannamker disclose processing a host of web content including images, animated images (including animated GIFs). Ex-1006, [0136]. Wannamaker also discloses using a temporal sequence with respect to such animated images. *See* [7d], [7e]. Moreover, both Avellan and Wannamaker disclose using a video streaming format (e.g., Ex-1003, 5:19-22 and Ex-1006, [0267]-[0273]) and generating a unicast stream (e.g., Ex-1003, 11:3-10 and Ex-1006, [0229-0232]). Ex-1007, ¶194. Therefore, this limitation would have been obvious to a POSITA in view of Wannamaker and Avellan because Wannamaker’s purpose is to allow user interaction so there must be unicast support (vs. broadcast) to allow a specific user to request specific content be unicast only to that user. *Id.* POSITA would have been motivated to combine Avellan with the teachings of Wannamaker because of the overlap in both systems. *Id.* Both Avellan and Wannamaker are directed to server-side software that obtains web-based content, renders it using a browser, and then packages it for delivery in a video frame format. *Id.* And because of this similarity, POSITAs would have had a reasonable expectation of success in combining Avellan’s system with Wannamaker’s disclosure of virtual machines. *Id.* Moreover, such a combination would not require an alteration to Wannamaker which already support unicast streaming of content. *Id.*

X. CONCLUSION

For the reasons set forth above, Petitioner has established a reasonable likelihood that all Challenged Claims are unpatentable. Accordingly, Petitioner respectfully requests institution of *inter partes* review and cancelation of the Challenged Claims.

Respectfully submitted,

Dated: December 20, 2024

/Masood Anjom/

Masood Anjom (Reg. No. 62,167)

Counsel for Petitioners,
STINGRAY GROUP INC. and STINGRAY
MUSIC USA, INC.

XI. CERTIFICATE OF WORD COUNT

Pursuant to 37 C.F.R. §42.24(d), Petitioner hereby certifies, in accordance with and reliance on the word count provided by the word-processing system used to prepare this petition, that the number of words in this paper is 13,864. Pursuant to 37 C.F.R. §42.24(d), this word count excludes the table of contents, table of authorities, mandatory notices under §42.8, certificate of service, certificate of word count, appendix of exhibits, and any claim listing.

Dated: December 20, 2024

/Masood Anjom/

Masood Anjom (Reg. No. 62,167)

Counsel for Petitioners,
STINGRAY GROUP INC. and STINGRAY
MUSIC USA, INC.

CERTIFICATE OF SERVICE

The undersigned certifies pursuant to 37 C.F.R. §§ 42.6(e) and 42.105(a) that on December 20, 2024, a true and correct copy of the foregoing “PETITION FOR INTER PARTES REVIEW” was served via Priority Mail Express on the Patent Owner at the following correspondence address of record as listed in the USPTO electronic filing system:

86000 - Fox Rothschild LLP / G. Nelson
212 Carnegie Center
Suite 400
Princeton, NJ 08540

and was served via electronic mail upon counsel of record for the Patent Owner in the litigation in the U.S. District Court for the Southern District of Florida:

Edwin A. Hernandez v. Stingray Digital Group, et al
Case No. 1:24-cv-21226-RAR

Matthew S. Yungwirth	msyungwirth@duanemorris.com
Alice E. Snedeker	aesnedeker@duanemorris.com
David C. Banker	dbanker@bushross.com
Celine Ohanian	COhanian@jmbm.com
Lena Streisand	LStreisand@jmbm.com
Stanley M. Gibson	SGibson@jmbm.com
Elio F. Martinez, Jr.	elio.martinez@gray-robinson.com
Francesca Russo-Di Stauro	francesca.russo@gray-robinson.com

Dated: December 20, 2024

/Masood Anjom/
Masood Anjom (Reg. No. 62,167)

Counsel for Petitioners,
STINGRAY GROUP INC. and STINGRAY
MUSIC USA, INC.