UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Stingray Group, Inc.
Petitioner
v.
Dr. Edwin A. Hernandez, Inc.
Patent Owner

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

DECLARATION OF DR. EDWIN A. HERNANDEZ-MONDRAGON

TABLE OF CONTENTS

I. INTRODUCTION....4

II. QUALIFICATIONS....7

III. LEGAL UNDERSTANDING....11

IV. PERSON OF ORDINARY SKILL IN THE ART....12

V. OVERVIEW OF THE '002 PATENT....13

A. Separating Rending, Obtaining, and Retrieving Allows Overlapping of Media Streams....15

B. The MediaPlug....17

C. Playlists and MIME Types....17

D. Video Encoding and Multimedia Containers (Encapsulation)....18

VI. CLAIM CONSTRUCTION....20

E. Content Provider....20

F. "Obtaining content corresponding to the plurality of multimedia items from at least one source"....21

G. "One media Stream for Playback"....21

H. Web Page....21

I. Render a Web Page by a browser using the content....22

Level of Ordinary Skill in the Art....22

Anticipation....23

Inherency....25

Obviousness....25

VII. SUMMARY OF OPINION....30

VIII. PERSON OF ORDINARY SKILL IN THE ART....34

IX. PATENTABILITY ANALYSIS FOR CHALLENGED CLAIMS 35

A. Ground 1- Claims 1-6, and 13 are not obvious under Avellan 38

B. Ground 2: Claims 1-6, 13 are not obvious under of Avellan, in view of Ma, and Fain. 64

C. Ground 3: Claim 7-9 and 13 are not obvious under Wannamaker 74

D. Ground 4 –Claims 10 is not obvious under Wannamaker in view of Avellan 100

X. CONCLUSION 102

XI. CERTIFICATION 102

I. INTRODUCTION

[01] My name is Dr. Edwin A. Hernandez-Mondragon. I understand that I am submitting a declaration for my own patent. I am offering technical opinions in connection with the *Inter Partes* Review ("IPR") proceeding in the United States Patent and Trademark Office for U.S. Patent No. 10,524,002 ("the '002 Patent"), and I have reviewed the references listed in Petitioner's Exhibit List. I make this declaration based on my personal knowledge. I am over the age of 21 and am competent to make this declaration. I have submitted reports and have been deposed in several IPR proceedings.

[02] The statements herein include my opinions and the bases for those opinions, which relate to at least the following documents and references of the pending IPR Petition, which I have reviewed and considered:

- U.S. Patent No. 10,524,002 (Ex. 1001)
- Claim Construction Briefs filed as part of S.D. of Florida litigation (Ex2052)
- Deposition of Mr. Lipoff, Day #1 and Day #2 (EX2053), Exhibit #3 (EX2054) and (EX2055)
- Petition for the '002 patent

- The exhibits referenced by Lipoff and attached to the Petition:

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-1008	Munford Decl.	Declaration of June Munford including *Curriculum Vitae*
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

And,

Table 2. Additional Exhibits Considered.

Exhibit No.	Description
EX2040	MPEG Handbook (Book Segment)
EX2041	Flash ActionScript (Book)
EX2042	H.264 and MPEG-4 A Video Compression, Video Encoding Multimedia (Book)
EX2043	War of Mobile Browsers (Dr. Edwin Hernandez' Paper, IEEE Pervasive Computing)
EX2044	Motorola OM1000 Manual
EX2045	HTML5 Black Book

EX2046	The Linux Programming Interface
EX2047	Linux Man Page for Errors
EX2048	RFC2326 – Real Time Streaming Protocol
EX2049	I-Node POSIX Paper
EX2050	Cloud Computing Virtualization a Comprehensive Survey
EX2051	CV of Dr. Edwin A. Hernandez – June 2025
EX2052	Case 1:24-cv-21226-RAR Document 188
EX2053	Stuart Lipoff Deposition Transcript on September 2nd, 2025
EX2054	Exhibit #3 Presented to Stuart Lipoff
EX2055	Stuart Lipoff Deposition Transcript on September 3rd
EX2056	RFC8216 – HTTP Live Streaming
EX2057	Containers and Cloud from LXC to Docker
EX2058	Avellan's Patent – File Wrapper
EX2059	Motorola OM1000 Pamphlet
EX2060	Analysis of Dalvik-VM Publication
EX2061	Liberate Connect Server
EX2062	Liberate Document TV Navigator
EX2063	RFC2112 – MIME Types

[03] Although I am being not being compensated for my regular hourly rate, I don't count with resources to pay for expert services or attorney's fees that range $200K to $250K per case. Hence, I am submitting this declaration to respond to petitioner.

[04] I am the Patent Owner of the '002 Patent and founder of EGLA CORP, a corporation owned by myself, and my elder parents, Dr. Alcides Hernandez and Mrs. Reina Gladys Hernandez. I had a verbal licensing agreement with my parents which the PTAB court requested to memorialize in written form which was submitted to this court and submitted by Robert Drolet

II. QUALIFICATIONS

[05] I received a B.S. from Costa Rica Institute of Technology in computer engineering, a M.S. from the University of Florida with an emphasis in electrical and computer engineering, and a Ph.D. from the University of Florida with an emphasis on computer engineering. A copy of my *curriculum vitae*, which includes a more detailed summary of my background, experience, patents, and publications, is attached as EX2051.

[06] I am an expert consultant in the fields of cable television systems and broadcasting, multimedia streaming, mobile devices and systems, air-interface and Long-Term Evolution (LTE), cloud storage and data synchronization, wireless communications, block-chain, power management, personal area networking, and smart phones and wireless embedded software development.

[07] I founded COMPUNET in 1997 and was the lead engineer from 1997 to 2009. While at COMPUNET, I was a lead developer for authentication services, security services, web services, and networking configuration services.

[08] I worked for Microsoft from 2001 to 2003. As a Technical Program Manager, I was responsible for driving architecture, design, test automation, and security analysis for Bluetooth Personal Area Networking (PAN). I also drove testing over networking protocols, such as IPv4 networks and IPv6 networks.

[09] I worked for Motorola, Inc. from 2003 to 2010. As a Principal Staff Software Engineer, I was responsible for application development for Google and Android platforms. I participated in kernel-level prototyping, data support, and digital rights management (DRM).

[10] Starting in 2010, I founded EGLA Communications. At EGLA Communications, I created MEVIA applications, such as Clout to Cable. MEVIA is a "software-as-a service" and a cloud-based platform that enables "MEVIA Music," which is currently in operation in several countries including Brazil, Honduras, and the United States. Cloud to Cable is a patented platform that merges cloud and cable television systems and simplifies music and video distribution to different platforms. Cloud to Cable is servicing operators, such as CABLE COLOR in Honduras.

[11] As part of my experience in EGLA, I have worked for cable TV systems in multiple operators: CABLEVISION Mexico, Axtel TV, CLARO, Direct TV, SKY Brazil, and many others. Hence, my technical experience and training covers cable TV systems, STBs, video-on-demand (VOD) systems, and several broadcasting methodologies.

[12] Additionally, over my career, my research has involved aspects of network security, wireless communications, network and communications reliability, artificial intelligence, multimedia streaming, and software engineering.

[13] I am a named inventor on sixteen patents issued by the United States Patent and Trademark Office, including the following:

- U.S. Patent No. 7,564,810 – Method and System for Managing Power Consumption of a Network Interface Module in a Wireless Computing Device

- U.S. Patent No. 7,231,330 – Rapid Mobility Network Emulator Method and System

- U.S. Patent No. 7,697,508 – System, Apparatus, and Method for Proactive Allocation of Wireless Communication Resources

- U.S. Patent No. 8,213,417 – System, Apparatus, and Method for Proactive Allocation of Wireless Communication Resources

- U.S. Patent No. 7,269,388 – Bluetooth PAN Driver

- U.S. Patent No. 8,788,715 – Rules-based Network Selection Across Multiple Media

- U.S. Patent No. 7,996,505 – Rules-based Network Selection Across Multiple Media

- U.S. Patent No. 8,024,487 – Smart Scan for Bluetooth PAN Services

- U.S. Patent No. 8,707,337 – Java-based Push to Talk

- U.S. Patent No. 7,331,793 – Magnetic Connector

- U.S. Patent No. 10,123,074, 10,524,002, 11,140,441, and 12,075, – Method, System, and Apparatus for Multimedia Content Delivery to Cable TV and Satellite Operators, with EP3238457A1.

- US Pat. No, 11,441,773 System, Method, and Apparatus For Virtualized Operations for Biohazard Waste Destruction with Augmented Reality and Voice Commands

- US Pat. No. 12,346, 399. Method and System for Web Interaction with Objects and Remote Display Technologies

- And, several pending applications.

[14] I am not an attorney and offer no legal opinions, but in my work, I have had experience studying and analyzing patents and patent claims from the perspective of a person skilled in the art, and I am a named inventor on several patents.

III. LEGAL UNDERSTANDING

[15] My understanding of the law regarding patent validity is based on my prior work on patents and patent matters and based on the information presented in this section. In formulating my opinions and conclusions in this case, I have been provided with an understanding of the prevailing principles of U.S. patent law that govern the issues of patent claim interpretation and validity. As a result, I understand the following principles of U.S. patent law and have applied these principles in analyzing the allegations of invalidity of the claims presented in the IPR Petition and in forming my opinions.

[16] I understand that it is a basic principle of patent law that assessing the validity of a patent claim involves a two-step analysis. In the first step, the claim language must be properly construed to determine its scope and meaning. In the second step,

the claim as properly construed must be compared to the alleged prior art to determine whether the claim is valid.

[17] As explained herein, my analysis of the validity of the ‘002 Patent will be undertaken from the perspective of what would have been known or understood by one of ordinary skill in the art relevant to the patent in question when the inventors of the patent conceived of and reduced the claimed inventions to practice. Whether any of the claims of the ‘002 Patent is anticipated or rendered obvious by systems and/or methods alleged by Petitioner to have been publicly disclosed, invented by another and/or in public use prior to the invention date is thus determined based on an understanding of a person of ordinary skill in the relevant art.

IV. PERSON OF ORDINARY SKILL IN THE ART

[18] A POSITA with respect to the ‘002 Patent would have education, experience, and training commensurate with a person with a bachelor’s degree in electrical engineering or computer science, and would have knowledge of remote desktop protocols, access controls, networking protocols, and communications, including TCP/IP-based standards, software design, distributed systems, and network equipment configuration. Based on my experience, education, and training, I have an understanding and knowledge of these capabilities and have been involved with these technologies myself.

V. OVERVIEW OF THE '002 PATENT

[19] The '002 Patent describes a system and method for providing that uses web resources such as a "Web Page" or a "Web Service" to create "User Interfaces" using the web to be part of broadcast channels.

[20] The '002 Patent describes how by using a browser or a rendering engine, the rendered "web pages" or "Custom UI" images are saved at different intervals of time.

[21] Separately, a plurality of multimedia items is retrieved from the cloud or another web source. This multimedia content ***is not part*** of the "webpage" rendering process, and as a result a media stream is created to be used by a client's device, e.g. a Set Top Box, a Smart TV, or any other device.

[22] One novel aspect of the '002, is the introduction of the term "Custom UI" or "Custom User Interface." The "Custom UI" is configurable via a particular "URL" pointed to a domain name or website, or what normally is called a web "link." See *Id.* at 5:54-63

> In some configurations a customizable user interface (UI) can be added to the output video MPEG Transport stream. As such, if the "Custom UI" variable is then true and valid, **325**, a link or URL **330** is loaded into the Mediaplug configuration that can be used when the video interface is created. The custom video interface can be created and fetched as needed or it can be created in real-time for a particular channel or for all the channels. The custom user interface variable can be uniquely configured per channel in the Mediaplug. The mediaplug can also monitor **340** all of

[23] In the ‘002 patent in one embodiment discloses the use of a browser to retrieve the “Custom UI” from a web-page located at a particular “URL” or “Universal Resource Locator.” In other embodiments, the “Custom UI” is retrieved from a web-service with JSON or XML definitions.

[24] Consequently, the ‘002 patent teaches that when a “Custom UI” or “Custom User Interface” (See *Id.* 6:63-56 and 7:1-6) is in use, the URL is used as “input” to the rendering process and such rendering follows standard HTTP protocols and a web browser that captures screens and turns them into media streams using software tools such as FFMPEG.

[25] The ‘002 takes advantage is to use the web and cloud. The methods in the patent disclose a novel solution where traditional multimedia playlists and web pages did not operate before. This enriched environment mixes media streams for playback, web pages, and multimedia items. The media stream resulting is delivered to content providers in the same format for digital “Satellite” or “Cable TV” Multiplexer technology.

[26] Additionally, the ‘002 patent describes how multimedia items that are part of a “playlist” can be loaded in to a device, called the MediaPlug. The Mediaplug

supports web protocols such as HTTP and "rsync" that are used to obtain multimedia items securely (*Id.* at 4:3-12).

> The cloud **101** service files can be accessible via the internet **105** and can be stored in a cloud-based file system or cluster file system (e.g. GlusterFS or Hadoop, Amazon S3). These files can be multimedia files that include, as non-limiting examples, MPEG-4 Video files with H.264 encoded video and AAC encoded audio or MPEG-layer 3 audio as well (MPEG2 Video could also be used). These files can be accessed using a secured transport protocol such as, but not limited to, SSL or a synchronization tool as "rsync," commonly found in Linux/MacOS environments. The com-

Id. at 4:3-12

A. Separating Rendering, Obtaining, and Retrieving Allows Overlapping of Media Streams

[27] Another novel aspect of the invention is that given the steps followed, web pages can be used to "overlap" over a media stream with videos, as indicated by the '002 patent at 7:4-6.

> **7**
>
> 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. If a customizable UI is part of the configuration **518**, rendered images from a web service can be used to create overlapping video or replace any video in the video feed file. Likewise, this video or customized screen can be used when no video at all is defined in the multimedia file, e.g. Audio-only file. The

'74 at 7:1-9

[28] Clearly, a web page on itself can contain a reference to a "video," images, text, audio, etc.. However, the benefits of separating requests for media streams,

multimedia items and a rendering of a web page is that such "rendered web page" can be positioned anywhere on the screen. The rendered web page can "overlap" a video from a different stream. As an example, a "stock ticker" is a web page, that can be overlapped over a "Bloomberg Video" stream, for example:

The original playlist video or feed is as follows:



A web-page contains a news or stock ticker:



The result of using the '002 is a broadcast media stream with the "rendered web page" as shown here:



B. The MediaPlug

[29] As explained before, the methods and systems disclosed by the ‘002 patent are stored in the “MediaPlug” device (See *Id.* at FIG.1 Element 110 and FIG. 2).

[30] The ‘002 Mediaplug can be implemented in a server or set of servers configured to provide media streams to provide media streams to operators as they used to do via the satellite.

C. Playlists and MIME Types

[31] Further, the ‘002 patent teaches that such playlist is stored as part of an internal web server are processed in response to an HTTP Request where the web server changes the HTTP “header” information of the response to be announced to a requestor, e.g. a Web client that, a playlist is in use. As specified in the RFC 2112 [1], (EX2063) the protocol provides a “MIME-like message” containing information on the request made.

[1] https://datatracker.ietf.org/doc/html/rfc2112

[32] Several common mime types include "text/html", "text/xml" and "image/png," In the case of "text/html" and "text/xml" a browser retrieving such request will first "parse it" and then "process" all "tags" in XML and HTML. The browser then looks for additional links that will be "parsed" and rendered as the response HTML indicates, requiring multiple requests to download images, text, fonts, JavaScript, or Ads (e.g. Google Ads)

[33] The 074 patent teaches that when the content is a multimedia playlist, the web server will return as "Content type", or MIME Type: "audio/x-mpegurl" as shown in *Id.* at 6:22-27.

completed **415**. At **420**, the caching server can generate a "playlist file" of mime type "audio/x-mpegurl" which will be interpreted as a multimedia playlist file when requested. Finally, as the playlist usually may not vary for a period of time until a new request is received, the playlist can be kept in the local memory caching service **425**.

Id. at 6:22-27

[34] Such playlist includes a list of files encapsulated in some type of container which can be but not limited to: MPEG-TS and MPEG-4 Part 14 as shown in *Id.* at 6.8-10.

D. Video Encoding and Multimedia Containers (Encapsulation)

[35] Multimedia container formats MUST not be confused with a video encoding formats, as some containers can use certain encoders, as shown in the '002 patent at 6:3-8, some sample of video encoders include H.264 and audio encoders such as

AC-3. Video and Audio Encoders use parameters such as: bitrates, screen resolutions, Group of Pictures, among many others.

[36] Therefore the '002 teaches that requesting the encapsulation "format" that includes encoded video and audio (See *Id.* at 6:39-40).

by the "Caching" unit. At step **505**, encoding flags or input parameters can be received which may include GOP (Group of Pictures) which is used by certain video encoders, resolution (e.g. 640×480) in pixels, bit rate in Kilobits per second, any encoding type of video (e.g. H.264), encoding type of audio (e.g. AC-3), duration of the video feed, etc. In addition, a "format" parameter can be received which refers to the output format or encapsulation, MPEG TS or MPEG4. The caching unit encodes audio and video, and can use the disk space in the Mediaplug. Hence in **508** the method

Id. at 6:33-43

[37] The main reason why encapsulation and encoding are different is because, encapsulation is a much faster process than encoding.

[38] Therefore, a POSITA will know that the MPEG-4 specification includes **video codecs** in various screen resolutions such as: MPEG-4 Part 10 or H.264/AVC, MPEG-H Part 2, and others that define **"encoding profiles"** that are supported by "video encoders" including H.264, MPEG2Video, and "audio encoders" such as AC-3, or MP3.

[39] Hence, the '002 specification discloses a solution compatible with HTML4/-5 and modern-day browsers with multimedia items and broadcast systems in use by Cable TV Providers (e.g. Multicast at *Id.* 10:6-10) or Smart TV systems with HTTP Live Streaming (e.g. HTTP Server with HTTP Live Streaming at *Id.* 8:52-55)

[40] In summary, the ‘002 teaches to use of web-based user interface that is rendered by a browser and a separate retrieval of a plurality of multimedia items to be used in the media stream for broadcast, what this means is that the plurality of multimedia items that are used by the ‘002 Patent.

VI. CLAIM CONSTRUCTION

[41] I understand that claim construction begins with the language of the claims, and the words of a claim are generally given their ordinary and customary meaning, which is the meaning the term would have to a POSITA at the time of the invention. I understand that although the prosecution history often lacks the clarity of the specification and thus is less useful for claim construction purposes, it is a source of intrinsic evidence that can inform the meaning of the claim language by demonstrating how the inventor understood the invention and whether the inventor limited the invention in the course of prosecution, making the claim scope narrower than it would otherwise be. I understand that while extrinsic evidence, such as expert testimony and dictionaries, may be useful in educating regarding the field of the invention or helping determine what a POSITA would understand claim terms to mean, extrinsic evidence in general is viewed as less reliable than intrinsic evidence.

E. Content Provider

[42] A “Content Provider” as the name says is an entity that receives content from multiple sources and generates at least “one media stream for playback”.

F. "Obtaining content corresponding to the plurality of multimedia items from at least one source"

[43] Under the plain and ordinary meaning, as one request to a source independently from "Rendering of a web-page"

G. "One media Stream for Playback"

[44] A " One Media Stream for Playback" is streaming content composed by audio and video frames, that can be played immediately.

H. Web Page

[45] Petitioner has already proposed in claim construction the term in District Court, and by no means includes equates a "media stream" is a "web page"

> In criticizing the proposed constructions' reference to HTML, Plaintiffs conflate two different concepts from the Asserted Patents. Specifically, Plaintiffs argue that the specification "teaches customizing a User Interface that is designed using Hypertext Markup Language (e.g. HTML) ***or a web service***," which the patents describe as using "XML (Extensible Markup Language) to respond to a custom UI request." *See* Dkt. 193 at 3 (emphasis added). As argued by Plaintiffs, "[b]ecause the specification teaches both HTML and XML markup language, Defendants' proposed construction would improperly exclude this preferred embodiment." *Id.*
>
> However, the process of designing a custom HTML user interface (even if done using XML) is entirely distinct from the process of "rendering a web page by a browser" that is at issue.

at Dkt. 198, Pg. 4 of 16, Case 1:24-cv-21226-RAR

I. Render a Web Page by a browser using the content

[46] The rendering process implies that the "content" from the previous step is already available

[47] "Render a Web Page" is described by the '002 patent at *Id.* 30-34, as a "Custom UI" or "Web Page" is used to capture a screen and creating an MPEG Transport Stream from calling a URL.

> Those screens are generated by calling a URL where the web-service is associated and creating a video file with the input audio and the captured screens. The output format can be an MPEG Transport Stream file that can be retrievable or streamed to the multicasting embodiment.

[48] Therefore, rendering of a web page is nothing but retrieving web resources and creating a displayable screen with the items retrieved from the web.

Level of Ordinary Skill in the Art

[49] The level of ordinary skill in the art is based on factors such as the educational level of the inventor, the educational level of those who work in the industry, and the sophistication of technology involved, in addition to the type of problems encountered in the art, prior art solutions to those problems, and the rapidity with which innovations are made in the particular technology including software engineering practices..

Anticipation

[50] I understand that to anticipate a patent Claim under 35 U.S.C. § 102, a single asserted prior art reference must disclose each and every element of the claimed invention, either explicitly or inherently to a person of ordinary skill in the art. There must be no difference between the claimed invention and the disclosure of the alleged prior art reference as viewed from the perspective of the person of ordinary skill in the art.

[51] It is my understanding that a patent is anticipated under 35 U.S.C. § 102 if (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for patent, or (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of the application for patent in the United States. The priority of invention goes to the first party to reduce an invention to practice unless the other party can show that it was the first to conceive the invention and that it exercised reasonable diligence in later reducing that invention to practice. It is my understanding that for prior art to be known under § 102(a), it must be publicly accessible and it must be sufficient to enable one with ordinary skill in the art to practice the invention. Public use under 35 U.S.C. § 102(b) includes any use of the

claimed invention by a person other than the inventor who is under no limitation, restriction or obligation of frequency of use.

[52] It is my understanding that to establish anticipation under § 102(a) on the basis of a printed publication, a party must demonstrate where in the publication each and every limitation of the claimed invention is found. Thus, each and every limitation must be found either expressly or inherently in a single prior art reference. A limitation is inherent if it is necessarily present in the prior art. The identical invention must be shown in as complete detail as is contained in the claim.

[53] Furthermore, it is my understanding that a reference must clearly and unequivocally disclose the claimed invention or direct those skilled in the art to the claimed invention without any need for picking, choosing, and combining various disclosures not directly related to each other by the teachings of the cited reference.

[54] Similarly, it is my understanding that, under § 102(b), a device that is used in public or sold may render a patent invalid on the basis of anticipation only if the device includes every limitation of the later claimed invention.

[55] Also, I understand that in order for a reference to be an anticipating reference, it must describe the claimed subject matter with sufficient clarity to establish that the subject matter existed and that its existence was recognized by persons of ordinary skill in the field of the invention.

Inherency

[56] I understand that in order to establish that an element of a claim is "inherent" in the disclosure of an asserted prior art reference, the extrinsic evidence (or the evidence outside the four corners of the asserted prior art reference) must make clear that the missing element is the inevitable outcome of the process and/or thing that is explicitly described in the asserted prior art reference and that it would be recognized as necessarily present by persons of ordinary skill in the relevant field. I understand inherency may not be established by mere probabilities or possibilities. In other words, the mere fact that a certain thing may result from a given set of circumstances is not sufficient.

Obviousness

[57] I understand that even though a prior art reference does not fully anticipate a claim of a patent, a claim may, nonetheless, be rendered obvious to one of ordinary skill in the art if the differences between the subject matter set forth in the patent claim and the prior art are such that the subject matter as a whole of the claim would have been obvious at the time the claimed invention was made. In addition, I understand that obviousness is a determination of law based on various underlying determinations of fact. In particular, these underlying factual determinations include (1) the scope and content of the prior art; (2) the level of ordinary skill in the art at the time the claimed invention was made; (3) the differences between the claimed

invention and the prior art; and (4) the extent of any secondary conditions of non-obviousness. I understand that if a claim element is completely missing from each of reference of a combination, the claim is not obvious in view of that combination.

[58] I understand that secondary considerations are any considerations other than the first three enumerated above that tend to show that the claimed subject matter would not have been obvious. Such evidence may include the following:

- **Commercial success**: An invention that is commercially successful is unlikely to have been obvious because it otherwise would have been invented by others earlier.

- **Copying**: Copying a solution, rather than inventing a different solution, shows that the patented solution would not have been obvious.

- **Long-standing problem or need**: A persistent problem or need in the art that went unresolved clearly implies that the solution could not have been obvious.

- **Prior failure**: The failure of others to come up with a solution shows that the solution could not have been obvious.

- **Licensing**: When industry players would rather take a license than try to come up with an alternative solution, those players acknowledge that the solution was not obvious.

- **Praise by others**: When those of ordinary skill, aficionados, and the defendants themselves praise the claimed invention on the merits of the invention it could not have been obvious.

- **Teaching away**: When those of ordinary skill in the art facing the same problem as the inventor looked in different directions than the inventor, and when following their lead would lead further away from the patented solution, that is a significant indication of non-obviousness. By the same token, when the charge in the "wrong" direction is led by those who are well-resourced, well placed, or of higher-than-average skill, it is an even stronger indication of non-obviousness when their teachings lead away.

- **Unexpected Results or Industry Skepticism**: When experts, aficionados, and the defendants themselves expressly or implicitly acknowledge that the claimed technology was unexpected or expressed skepticism in it, that is strong evidence that the improvements could not have been obvious.

[59] To ascertain the scope and content of the prior art, it is necessary to first examine the field of the inventor's endeavor and the particular problem with which the inventor was involved at the time the invention was made. Moreover, a determination of obviousness cannot be based on the hindsight combination of components selectively culled from the prior art to fit the parameters of the claimed invention. Instead, I understand that: in order to render a patent claim invalid as being obvious from a combination of references, there must be some evidence within the prior art as a whole to suggest the desirability, and thus the obviousness, of making the combination in a way that would produce the claimed invention. In addition, I understand that in order to find a patent claim invalid for obviousness, there must be a finding that each element in each limitation of the patent claim is disclosed or taught by the asserted combination of prior art references or elsewhere in the relevant prior art. I further understand that in making a combination, the principle of operation of a reference should not be changed and that the prior art cannot be rendered unsuitable for its intended purpose.

[60] I understand that a patent composed of several elements is not proved obvious merely by demonstrating that each of its elements was, independently, known in the prior art. This is so because inventions in most, if not all, instances rely upon building blocks long since discovered, and claimed discoveries almost of necessity will be combinations of what, in some sense, is already known. I further understand that

most inventions arise from a combination of old elements and each element may often be found in the prior art. However, mere identification in the prior art of each element is insufficient to defeat the patentability of the combined subject matter as a whole. Rather, to establish a case of obviousness based on a combination of elements disclosed in the prior art, an articulation must be made on the basis by which it would have been obvious to make the claimed invention.

[61] I understand that in making combinations of references it is important to avoid hindsight and it can be important to find a reason to make a particular combination. Care must be taken to avoid the temptation to read into the prior art the teachings of the invention at issue and one must guard against slipping into the use of hindsight when considering the issue of obviousness.

[62] Particularly, one must avoid simply taking the inventor's disclosure as a blueprint for piecing together the prior art to defeat patentability – this is the essence of hindsight.

[63] I understand that when the prior art teaches away from combining certain known elements, the discovery of a successful means of combining them is more likely to be nonobvious, and evidence rebutting a case of obviousness can include evidence that the prior art teaches away from the claimed invention in any material respect. I understand a reference may be said to teach away when a person of ordinary skill, upon reading the reference, would be discouraged from following the

path set out in the reference, or would be led in a direction divergent from the path that was taken by the inventor. I understand that general skepticism of those in the art – not amounting to teaching away – is also relevant and persuasive evidence of nonobviousness. In effect, teaching away is a more pointed and probative form of skepticism expressed in the prior art. In either case, the presence of either of these indications gives insight into the question of obviousness.

[64] I understand a reference qualifies as prior art for an obviousness determination under §103 only when it is analogous to the claimed invention. Two separate tests define the scope of analogous prior art: (1) whether the art is from the same field of endeavor, regardless of the problem addressed and, (2) if the reference is not within the field of the inventor's endeavor, whether the reference still is reasonably pertinent to the particular problem with which the inventor is involved.

VII. SUMMARY OF OPINION

[65] The Petition sets forth four grounds:

Table 1. Grounds of Invalidity

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

[66] Claims 1-13 are the challenged claims. Claims 1, 4, 7, and 13 are independent claims the other claims are dependent.

[67] Mr. Lipoff testified that he relied on the petition as part of his analysis and did not conduct an independent analysis for all the claims (Declaration Pg. 144, lines 2-6, Pg. 116, lines 15-20, 23-25, Pg. 117, 1-8 and 19-23)

[68] Therefore, Lipoff *only* provided partial analysis and relied on the Petition for many of the claims.

[69] Hence, the Petition and Lipoff failed to provide a convincing argument for invalidity.

[70] For, Ground 1 for Claims 1-6, and 13 relies on Avellan, Lipoff and Petition fails to show convincingly Ground 1 anticipates the '002 Patent.

[71] Avellan relies on "images" captured from "a web page," The Petition claims shows that a Gateway server works with "image captured" from a web page, that Avellan turns into compressed files with the "Web to HD Video Protocol." If a web page is captured from a "media web page", such "CNN, a Live Stream, Youtube, etc," Petitioner calls it a "media stream"

[72] Petitioner argues that the captured webpage also become a "one media stream for playback" and a "plurality of multimedia items." The '002 patent teaches that "retrieving" and "obtaining" are separate from "rendering" hence are not the same.

Claim 8 in the ‘074 Patent for example, shows that upon “detecting a change on a web page” simply retrieves the “web page again” using the already obtained “plurality of multimedia items” from a separate process.

[73] In contrast, Avellan can only capture images from a web page. If the web page has a video or audio, assuming holds a web player with an autoplayed, function the webpage will be captured otherwise it won’t. In contrast, the ‘002 Patent handles this gracefully by separating, web captured from “obtaining” multimedia items and media streams.

[74] By separating these efforts, the ‘002 specification teaches that “the rendered images from a web service can be used to create overlapping video” or “replace any video in the video feed file,” or even “be used when no video at all Is defined in the multimedia file, e.g. Audio-only file” (‘Ex1001, 7:6-11)

[75] Hence, the “rendered images” used as “User Interfaces” must be obtained separate from the “plurality of multimedia items” such that “rendered images from the web service” can be processed separately as overlapping any content on the screen (Ex1001, 7:6-11)

> with HTML4/5 or any future HTML version. If a customizable UT is part of the configuration **518**, rendered images from a web service can be used to create overlapping video or replace any video in the video feed file. Likewise, this video or customized screen can be used when no video at all is defined in the multimedia file, e.g. Audio-only file. The

Ex1001, 7:6-11.

[76] Therefore, equating "screen captures" of a web page that shows a "video" or "audio" to be a "one media stream for playback" and a "plurality of multimedia items" is an error.

[77] Besides that, Avellan operates with "compressed video files" derived from several images captured from "web pages" that need to be decompressed for display or "may be immediately displayed (once decompressed at computer 104) (Ex1002, 7:57), as those files are accompanied by a "Tag file" *Id.* 5:49-52

[78] Moreover, Avellan cites an unknown protocol called "Web to HD Video Protocol" and Petitioner makes several assumptions just to match the claim language in the patented invention.

[79] Ground 2 relies on Avellan, Ma, and Fain for Clams 1-6 and 13 relies on Avellan with all the issues with Avellan, Fain is a Java Programming Book, suggesting that if a "web page has a java applet or flash," then virtualization can

[80] Ground 3 relies on Wannamaker for Claims 7-9, 11-12, and petitioner points to the MMS function at Wannamaker for the multicast unit however, as attached the MMS unit relies on MMS_OC1000 device from Motorola that only "receives" multicast, cannot generate them. Besides that, Wannamaker feeds the outputs of the MMS device into the QAM Upconverter, and does not cover the claims,

[81] Ground 4 relies on Wannamaker and Avellan for Claim 13, clearly, besides all Avellan's issues and Wannamaker competing technologies with Avellan. Wannaaker is a hardware technology and a POSITA will not be motivated to combine it, besides Wannamaker and Avellan attempt to solve the same problem, displaying web pages a) Avellan low-bandwidth users, and b) Set Top Boxes with limited processing capacity, however both system fail to operate for multimer reasons.

[82] In sum, Avellan does not show how "at least one stream," and can only work at best can act as an intermediatory entity for "…HLS, HTTP Playlist, or RTSP…" stream that reside on web site, requiring also of a web player and "sending a click to the –play—button" that the '002 can directly tackle for better results.

VIII. PERSON OF ORDINARY SKILL IN THE ART

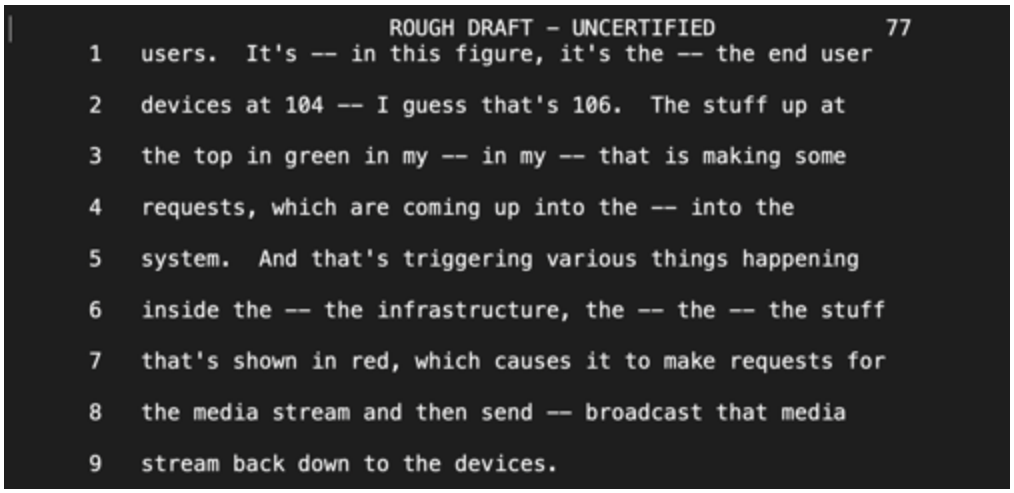
[83] A POSITA with respect to the '002 Patent would have education, experience, and training commensurate with a person with a bachelor's degree in electrical engineering or computer science, and would have knowledge of remote desktop protocols, access controls, networking protocols, and communications, including TCP/IP-based standards, software design, distributed systems, and network equipment configuration. Based on my experience, education, and training, I have an understanding and knowledge of these capabilities and have been involved with these technologies myself.

IX. PATENTABILITY ANALYSIS FOR CHALLENGED CLAIMS

[84] It is my opinion that the references cited by Petitioner fail to disclose or each and every element recited in the challenged claims, and therefore fail to render obvious the challenged claims. It is further my opinion that it would not be obvious to POSITA to modify the references relied upon by Petitioner with the missing elements in such a way that would result in the system and methods recited in the claims of the '002 Patent.

[85] The "Anticipation" arguments disclosed by Petitioner fall into at least into these categories:

a. Web Page is the same as a one media stream, hence when a web page is retrieved, all media stream claims are met.



Lipoff Declaration at Pg. 77:1-9

b. The mechanisms behind: obtaining, retrieving, rendering, and assembling are all the same.

c. For the Avellan reference, "content provider" it can take different shapes in some cases is the "provider of content" in others is the "recipient." As shown at Ex1002¶93 at FIG 1, for some claims is the "Red Arrow" for other claims it the "Green Arrow"

d. Transcoding of HTML into CPML/MML is the same as rendering and creating "screen captures"

e. A Multimedia Server is any entity that has a word "MPEG" attached to it, and that can receive or transmit information

f. Any caching system is the "Caching Unit" but sometimes it the System.

g. The Web-to-HD Video protocol is "well-known," although there is not a single reference about it, a standard or documentation.

[86] For at least the reasons presented below, I disagree with Petition statements and Mr. Lipoff.

- First for Ground 1, Claims 1-6, and 13. Avellan relies on the Petitioner and points to the "a web page" being the same as "one media stream for playback," when these are two different concepts explained in the

intrinsic evidence, as well as in the technical dictionary definitions, additionally Petitioner requires that the "plurality of multimedia items" to be retrieved thru a web page rendering, when this process is separate and different, as the claim describes for example "one media stream for playback", and separately "rendering of a web page."

- Also, Petition relies in an obscure citation called "Web to HD Video Protocol" that makes enablement questionable.

- Second Ground 2, Claims 1-6 and 13 in view of Avellan, Ma, and Fain. Avellan and Ma are competing technologies, as Avellan will use Ma's HTTP Live Streaming, besides the issues with Ground 1. Ma's HTTP Live Streaming "media streams" by being screen captured or image captured by Avellan, this combination will cause a degraded user experience, for example a Live stream in 4K will be turned into "compressed video files" that now need to be "uncompressed" to be seen.

- Ground 3 Claims 7-9 and 10-11 under Wannamaker as a reference to invalidate However, Wannamaker relies on Transcoding to convert HTML into CPML/MML and FBM images. Wannamaker sends over MPEG this content to be processed by a proprietary TV Navigator

Application. Besides that, Wannamaker relies on an MMS hardware device by Motorola called OM1000 that simply can only receive multicast, not create them. Hence, this teaches away from the Claim 7. Given this contradiction and other deficiencies, Ground 3 is not met by Petitioner and all claims are valid.

- And, Ground 4, Claim 10 in view of Wannamaker and Avellan, as Wannamaker turns a web page via Transcoding into CPML/MML and images, Avellan turns a web page into image captures, such combination is not compatible and will require to competing models to collide.

A. Ground 1- Claims 1-6, and 13 are not obvious under Avellan

[87] I begin my analysis with the challenged Claim 1:

computer-implemented method comprising:

a. 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;

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

c. rendering a web page by a browser using the content;

d. 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;

e. assembling the at least one media stream using the temporal sequence of screen captures; and

f. providing the at least one media stream to the content provider for broadcast on the broadcast media channel;

g. wherein the at least one media stream corresponds to at least one of an HTTP live stream (HLS)[2], an HTTP playlist, and a Real-time Streaming Transport (RTSP[3]) stream; and

[2] RFC8216 – HTTP Live Streaming Protocol (EX2056)

[3] RFC236 Real-time Transport Streaming Protocol (EX2048)

h. wherein the web page is rendered in parallel in multiple threads...

[88] Claim 1, Element 1[a] that recites "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". My analysis starts with the "receiving, from a content provider, a request"

a. The request from the content provider must exist, for example as part of an IGMP request to join a multicast session. The '002 Patent describes at Ex1001,6-11 states that "In order to generate IGMP (Internet Group Management Protocol) and establish a multicast group (e.g. 239.0.0.1:8000)"

b. Or, a as another example, of "***a request***" from the content provider Id. 5:45-49 "The playlist and channel configuration can be obtained from the content provider"

[89] The next section of the claim reads "for at least one media stream for playback". The '002 Patent describes what this is:

a. First, Lipoff declared that the "Red Arrow" at Ex1002 ¶93 points to the "Content Provider"

b. Second, the "one media stream for playback on a broadcast media channel" is a media stream encoded for a particular bit-rate and resolution, and using a particular audio codec,

[90] Avellan does not disclose a "one media stream for playback", as the "one media stream" is defined by the '002 patent as:

a. A **media stream** that is "compatible with MPEG-TS at a pre-determined rate or bitrate" See Ex1001 at 1:35

b. A **media stream** can assemble "using each of the plurality of the multimedia items" *Id.* 2:60-61

c. The various multimedia items can be used 608 to assemble the **media stream** in a format corresponding with the request (e.g. HLS, HTTP/RTSP, RTSP Stream) *Id*. 8:14-16 or "HTTP Playlist" *Id.* 7: 47-51

[91] Avellan in contrast do not provide a "***one media stream for playback***" instead Avellan compresses the web page and then is "the user computer 104, where it may be **immediately displayed (once decompressed at the computer 104).**" Ex1004 at 7:56-58 (emphasis added)

[92] **The server 134 creates a compressed video format of the web page**. *Id.* 13:35-35 (emphasis added). The "compressed video file" is associated with a "TAG File" as *Id.* 11:36-37

[93] Hence, Avellan cannot disclose a "one media stream for playback," as such a media stream requires streaming over the "broadcast media," and does not need storage at the receiver end.

[94] Hence, Avellan does **NOT** disclose the "at least, one media streams for playback" due to each web page being first stored as **"compressed video file"** that is also tied to **"tag files"** that contains positioning information of all items on the web page. Hence, the compressed videos **are** treated as files to match the TAG File that is used for interactive clicking over items, for example:

a. As shown the traffic at FIG 1, element 120, 122, 124 at Ex1004 at 9:29-35 "While the bidirectional links include a Tag Communications Channel (Tcc) 124"

b. For instance, "The requested web page … imaged, compressed, tagged, cached, and broadcast over Bcc 120 to the user computer 104" Ex1004 13:63-67

c. Among other but as in *Id.* 10:63-37 "The CNN site can be stored in a local hard drive of the user's computers 104_i-104n if desired

This methodology conserves bandwidth because websites are transmitted as video images and the data is compressed, and transmitted to sever user's computers 104_i-104n

d. And, at *Id.* 11:42-47 "tag file which includes the domain name for the web page, the location where the action is taken (X pixels down and y pixels over) and the action taken (such as the movement or click of the mouse, the typing of a letter or command (tab, backspace, etc.) on the keyboard, etc.

[95] Moreover, Element 1[b] requires the process of "obtaining content corresponding to the plurality of multimedia items from at least one source offering the content in at least one first format"

[96] As Avellan "***compresses***" images from a web page, and such compression will change the resolution sizes, and any creation of a video will create a compressed version of the audio on the web page.

[97] Similarly, ***screen captures of a web page*** with a live feed, although visually may include images and graphics that are part of a web player, that is not going to provide a good user experience. In simple terms, getting "screen captures" of a live feed on a web page will be less than ideal reducing the frame rate from 30FPS to possibly lower, and possibly damaging surround sound and other audio stream

properties. Besides, most of the web pages do not show, initially the live stream as a "Full Screen" instead, simply will require the user to send a "click" over the TAG communications channel to the "Full Screen" button tag.

[98] Therefore, Element 1[b] that reads "*obtaining content corresponding to the plurality of multimedia items from at least one source*" implies that there could be multiple sources, not only, the worst option is to use the "web page," The '002 patent discloses this process as follows:

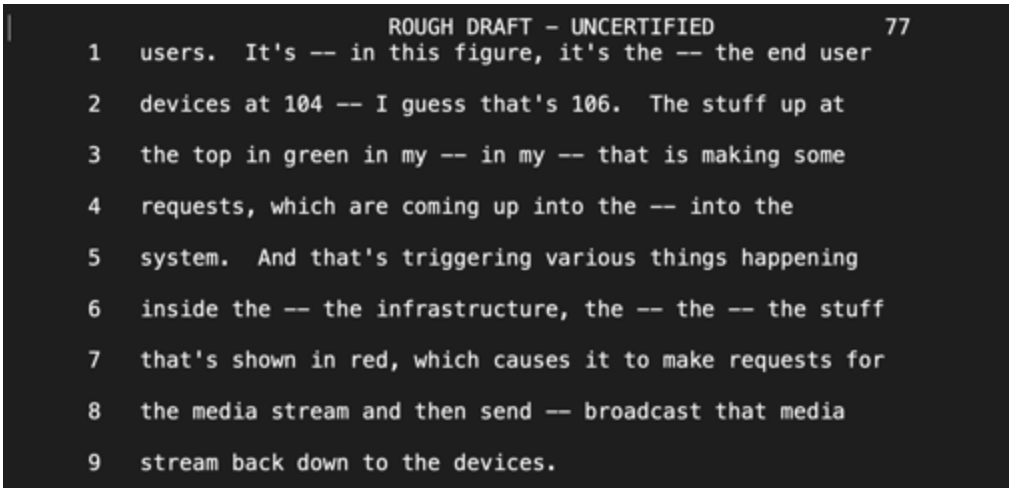
a. Via "rsync" protocol, See Ex1001 at 4:13

b. Via access to a "Cluster File System (e.g. Gluster FS)" See *Id*. at 4:5-8

c. By obtaining content from retrieved from the "media stream", e.g. a playlist.

[99] Hence, the only mechanism where the Petition and Lipoff points to Element 1[b] by a request made by the Gateway server to the **webpage.**

[100] In fact, Lipoff repetitively points to the web-page to be the media stream. As in Lipoff Deposition at 77:6-9 " – the infrastructure, the – the – the stuff that' shown in read, which causes it to make request for **the media stream**…" (**emphasis added**)



ROUGH DRAFT - UNCERTIFIED 77

users. It's -- in this figure, it's the -- the end user devices at 104 -- I guess that's 106. The stuff up at the top in green in my -- in my -- that is making some requests, which are coming up into the -- into the system. And that's triggering various things happening inside the -- the infrastructure, the -- the -- the stuff that's shown in red, which causes it to make requests for the media stream and then send -- broadcast that media stream back down to the devices.

Lipoff Declaration, EX2053 at 77:1-9

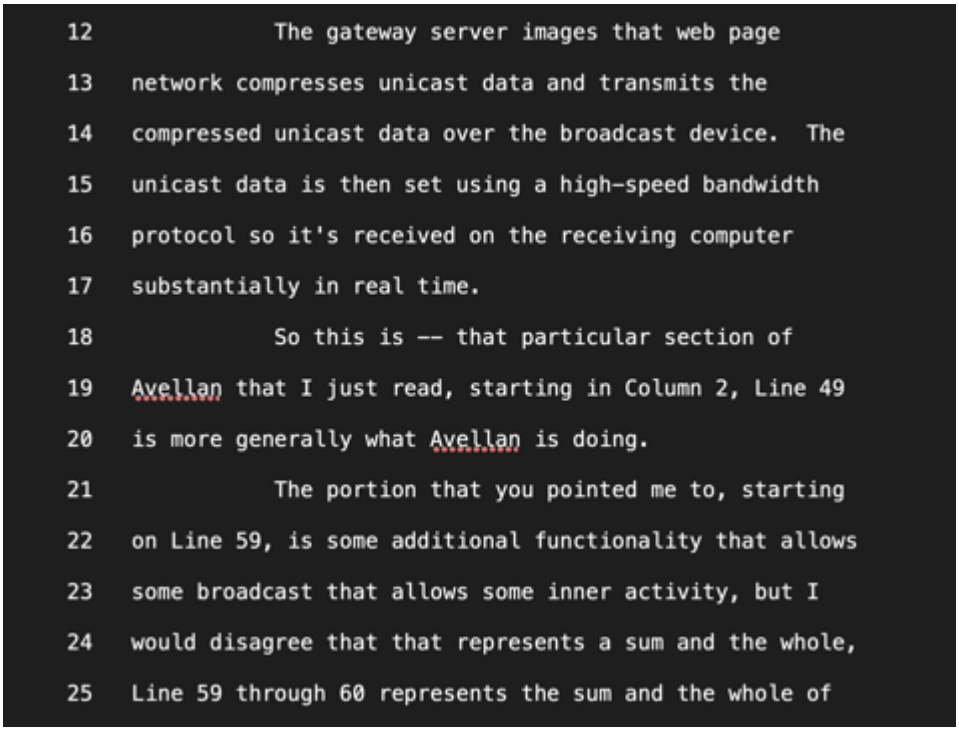
[101] Obviously, a "web page" is ***not*** a "media stream," hence the Claim 1[a] and [b] deals with multimedia content other than the web page, and Avellan can disclose these elements.

[102] It is clear that Avellan only obtains images or information when is part of the web-page that is "the gateway server 134 continuously captures images of a web page" Lipoff Deposition at 130:5-6.

[103] Hence, Avellan only discloses obtaining of the content when is part of a web page.

[104] Now, For Element 1[c] that involves "*rendering a web page by a browser using the content;*" As explained by Lipoff, in his view and Petition, when Avellan retrieves a web page also retrieves the media stream. Therefore, Lipoff points to

when "Gateway server images that web page network compresses …" downloads the content, as specified by the HTML (Lipoff Deposition Day 1 at Pg. 66:12-25

- 

```
12            The gateway server images that web page
13  network compresses unicast data and transmits the
14  compressed unicast data over the broadcast device.  The
15  unicast data is then set using a high-speed bandwidth
16  protocol so it's received on the receiving computer
17  substantially in real time.
18            So this is -- that particular section of
19  Avellan that I just read, starting in Column 2, Line 49
20  is more generally what Avellan is doing.
21            The portion that you pointed me to, starting
22  on Line 59, is some additional functionality that allows
23  some broadcast that allows some inner activity, but I
24  would disagree that that represents a sum and the whole,
25  Line 59 through 60 represents the sum and the whole of
```

  EX2051, 66: 12-25

[105] Hence, the Petition and Lipoff relies on the "*Web-to-HD video protocol*" presented by Avellan or the "Web to HD Video Protocol" which is referenced by EX1004 at 5:16-32 and briefly mentioned by Lipoff at ¶124 in Ex-1007.

[106] There is not a single publication online or in the Petition files that describes how "Web to HD Video Protocol" works. I searched Avellan's Patent File Wrapper (EX2058) without any light to any specification.

[107] Hence, Avellan disclosure does not presents a clear and concise explanation about "the protocol," except that such "video protocol" simply, converts web pages to compressed video and audio images (See Ex1003 at 5:16-32) and "sends" them.

> In a non-limiting aspect of the invention, 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). The protocol may be configured to compress the data into a compressed digital video format that can be sent to the user computer **104** for display in the browser **106**. Since HTTP is inefficient and slow, the web to HD video capture protocol improves the performance of satellite channels to be equal or better than 50 msec terrestrial high-capacity channels not accessed via HD Internet using the present invention. Though the web pages are preferably imaged, any suitable technique can be used to place the web pages in a format suitable for broadcasting to a large number of user computers **104**, such as converting the HTTP code to an image frame format or the like. Preferably, however, the technique is independent of the HTTP content, as in imaging.

EX1003 at 5:16-32

[108] Even if such "*Web to HD video protocol*" that "captures and transmits images/video of web page" *Id.* 5:17-18 exists it will fail to generate "media streams for playback" as well. Instead, the '002 patent teachings do not require of any "video protocol" or communication to achieve the functionality disclosed, as the patent itself describes how it works.

[109] Instead, the '002 provides a detail way to convert web pages into images and then to media streams as follows:

a. By using PhamtomJS a headless web-browser based on webkit or a tool such as QtCapture. Screens can be captured from a screen or rendered via a web browser, or a web-service with XML or JSON *Id.* at 6:63-57, 7:1-5, 7:18-22.

b. For example, a web-service can also be used to create a " Custom User Interface" with web resources or follow XML or JSON schema (*Id.* 6:61-57 and 7:1-5)

c. In general, a "Custom User Interface" is what is disclosed by the '002 patent.

d. Next, the captured images for the "Custom User Interface" are used as inputs for the "FFMPEG"[4] module or other software tool, that can convert a sequence of images into an MPEG Transport Stream, in *Id.* at 7:25-6, 7:39-40,

e. The MPEG Transport Stream created is then multiplexed using FFMPEG or VLC[5] with other multimedia files (e.g. Music), that were not part of the "Custom UI" process (See *Id.* at 7:39-40)

[4] https://ffmpeg.org/ffmpeg.html

[5] https://www.videolan.org/vlc/

f. The '002 patent explains that placing the MPEG Transport Stream files into an HTTP server that can be used as part of playlist or a media stream (See *Id.* at 7:55-57),

g. The generated MPEG Transport Stream is then a media stream for playback.

[110] Therefore, given the obscure nature of the "*Web to HD Video Protocol*" the things that are clear from Avellan are:

a. Avellan relies on compressed web page images captured into a video (as video snippets from the web page),

b. User computers received these compressed web page image files that required decompressing them for displaying (*Id.* at 7:57)

[111] Therefore, Element 1[d] that consists in "*assembling the at least one media stream using the temporal sequence of screen captures*;" is not present in Avellan. As simply speaking, the petition equates "**media streams**" with "**web pages",** hence a POSITA will not feel compelled to use Avellan to retrieve a "web page" as a "media stream" and assemble the "media stream" if it was already a one, as that ***circular rationale*** is simply flawed.

[112] Regarding Element 1[e] is also not met, as "providing the at least one media stream to the content provider for broadcast on the broadcast media channel," again

if the "Content Provider" is the same as the "providing the at least one media stream" such step would be also unnecessary as the in the assembling step the "media stream" was also provided, whereas in the '002patent, such step is needed as the content provider is the recipient of the "one media stream," and both the "one media stream" and the "content provider" are connected thru the "broadcast media channel"

[113] Therefore, by no means Claims 1[a] to 1[f] are disclosed by Avellan.

[114] My analysis considered the following statements made in the Petition:

a. Petition, 25 refers to a "web page" in the context of Avellan as part of the "browser software 106"

b. Avellan only renders web pages using a browser,

c. Lipoff at Ex1007 ¶122 references that a "live streaming feed" such as a "Hockey Game" but "omits" that is a web page with a live streaming feed,

d. Avellan describes at *Id.* 13:41-46 that when web page "rendered by the remote browsers such "Hockey Game" *Id.*¶122 will include a "video web player." As browser need a "video web player" to handle a live stream, such a player can use of Flash or Java Applets. These tools are used to create web pages (See Ex1003 13:41).

e. Hence, in Avellan, a user interacts with the browser using the TAG file. For example a "click" *Id.* 5:56-57

f. A POSITA will know that a "click" can be sent to a "play button" command on a website (*Id.* 5:33-36,)

g. The browser renders image captures from a web page (*Id.* 5:8)

h. The "remote virtual browser" then "converts that to a form suitable for display on the – the devices 104, 106" (See Lipoff Declaration 60:1-3).

[115] Therefore, for Element 1[g] "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

[116] Ex1007 ¶124 points to the "Web to HD Protocol" as if it includes HLS or HTTP Live Streaming protocol, this is improper and incorrect. The same reference used by IPR2025-0349 Ex-1009 (For Avellan in view of Mathews arguments), show HLS index files and how they operate.

[117] First, *Id.*¶126 mischaracterizes how HTTP Live Streaming works:

a. HTTP live streaming does not "send audio or video," HLS is described in the RFC8216 (EX2056)

b. First, HLS is an HTTP-based protocol consisting of two elements. A) The M3U8 playlist file, that is de fined at *Id.* at 10 "Playlist is a Media Playlist if all URI lines in the Playlist identify Media Segments." B) The segments, which are "MPEG Transport Stream" segments. *Id.* at Pg7

c. The web browser is change of retrieving first the Playlist, as an M3U8 link, parsing it. The playlist may contain a list of other M3U8 links that are available on the website for "Adaptive Bit Rate" features, which is called "Variant Stream" Id. at Pg. 5

d. In general, this list contains, let's say 10 segments of 10 seconds each. The M3U8 file or Media Playlist will contain:

```
#EXTM3U
#EXT-X-TARGETDURATION:10
#EXTINF:9.009,
http://media.example.com/first.ts
#EXTINF:9.009,
http://media.example.com/second.ts
#EXTINF:3.003,
http://media.example.com/third.ts
```

e. A Variant Playlist is then retrieved by the client's browser and processed to handle which bit-rate is done at the time.

f. Once the Playlists are known, the browser with a "Web Player" will be able to retrieve each MPEG Transport stream and use the appropriate video and audio codec that are needed to decode the media stream at a certain bit-rate.

[118] Hence, the explanation and flawed comparison of "Web to HD Video Protocol" with "HLS" is wrong and completely made up, as clearly, if it did, it will understand that HTTP Live Streaming is far superior and that, creating image captures from it would create an inferior user experience.

[119] RTSP or Real-time Streaming Protocol in contrast is different, as although it utilizes a protocol and "SDP" or the "Session Descriptor Protocol" as defined by RFC2326 (EX2048). In this specification describes that "*RTSP may use either an unreliable datagram protocol (UDP) (RFC768 [9]), a reliable datagram protocol (RDP, RFC 1151, not widely used [10]) or a reliable stream protocol such as TCP (RFC 793 [11]) as it implements application-level reliability*" at *Id*. Pg. 8.

[120] Cleary, neither HLS or RTSP "sends audio and video," the server simply replies to the Web Player in HLS to retrieve the segments is storing at the servers, and if any "sending is doing" is in response to an HTTP GET Request made by the browser's client. HTTP is a pull protocol not a "push" protocol, that "sends" packets.

[121] Regardless of erroneous statements made by Lipoff and the Petition, 30, Element 1[g] is not disclosed and can't be disclosed by Avellan's "Web-to-HD Video Protocol"

[122] For Element 1[h], "wherein, the web page is rendered in multiple threads," as shown above, Elements 1[c] "Rendering a web page using the content "what this means is that obtaining, and retrieving are also executed in different threads, besides the web page, and all components associated with such request.

[123] The '002 patent teaches that all processes can be run in parallel in threads as shown at Ex1001 at 6:46-48.

[124] Petition, 31 references "**Java**" as the programming language used by Gateway server. *First,* Java cannot be used for such purpose as all rendering engines including WebKit and Chrome (See EX2043) are written in C/C++; *Second,* the "Java" reference refers to the web site, that back in those days included "Java Applets" or Embedded Flash Object (Ex2041, Pg. 35) as described by Avellan at 13:37-41.

"The compressed video format may include all of the page's graphical information and data from different sources in a single video signal representing the image of the web page. The data may ***include Java, html, flash****, or other programs known to those of skill in the art."* **(emphasis added)**

[125] Besides, given the lack of specification and details on "*Web-to-HD Video Protocol*," Avellan fails to show that it can be executed in multiple threads, as it is unknown and whether such protocol potentially makes a request to another server (for example) hence it cannot be determined if it can be parallelizable.

[126] The Element 1[a] to 1[g] are clear and when 1[c] is rendered in multiple threads, all other processes can also be parallelizable as the Ex1001 teaches at 6:46-48.

[127] Therefore, Claim 1 is not anticipated by Avellan.

[128] Regarding Claim 2 language "The method of claim 1, "*The method according to claim 1, wherein the at least one media stream is interacted with MPEG two way control messages,*" an MPEG two-way system inserts at a "Program Identifier" (PID) a text (e.g. *URL or a link*) that occupies a program identifier other than video and audio. The '002 Patent at Ex1001 *Id.* 11:35-40 shows how the describes at The MPEG packet is extracted from the PID and passed to a browser in the receiving device. Avellan does not teach or show this behavior.

[129] The Transport Stream counts with three "Program Identifiers" or PIDs, video, audio, and text. The text is the MPEG-TS Control Messages (Two-way), DCII Text, or others. *Id.* FIG 9 shows how this is done at steps 902 and 905 shows how the MPEG Packet is inserted into the multicast stream.

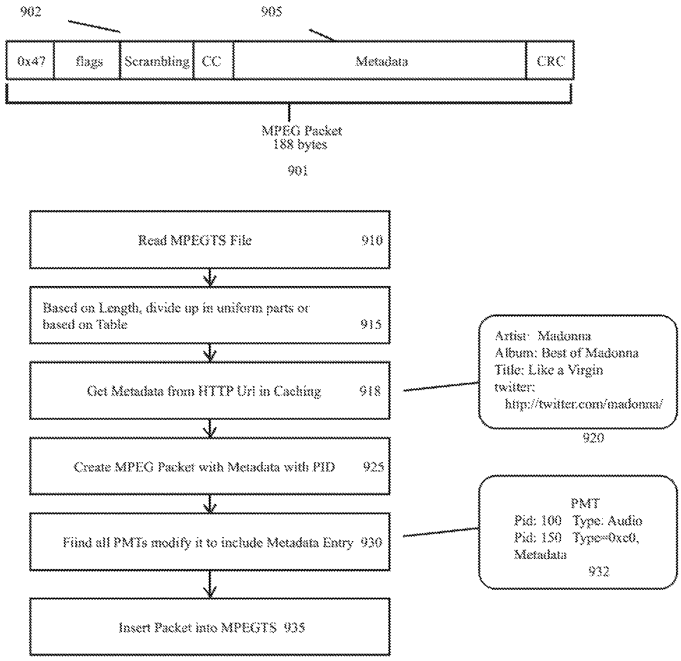


FIG. 9

‘002 Patent - Ex1001 at FIG 9

[130] A POSITA will now that as shown at FIG 9. Element 920 the metadata could be a URL, e.g. https://twitter.com/madonna/ or other web-site with tracking information for a particular use.

[131] Petition, 33 discusses something different to what the patent discloses and how MPEG Two-way systems work and operate.

[132] Hence, Claim 2 is not disclosed.

[133] Regarding Claim 3 language “The method according to claim 1, wherein MPEG messages are sent to a service provider via a unicast address,” for which

sending to a service provider using a unicast address such as 192.168.0.1 instead of a multicast address such as 239.0.0.1 (See Ex1001 4:31-33).

[134] For example, a user's set top box with IP address 192.168.100.100 and Port 1000 may receive a media stream only directed to that IP Address only, using a unicast packet.

[135] What Claim 3 shows is that using FFMPEG or other tools, it is possible to have a "udp sink" for example, where messages from the "MPEG Transport Stream" are sent to a "unicast" address. For example, "ffmpeg" can send a stream to udp://192.168.10.10:1000, which is a ***unicast address***. The recipient opens a TCP/IP or UDP/IP socket to receive the stream, Petition is ***silent*** on this matter. .

[136] Regarding Claim 4 language: A computer-implemented method, comprising:

a. 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;

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

c. rendering a web page by a browser using the content;

d. 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;

e. assembling the at least one media stream using the temporal sequence of screen captures; and

f. providing the at least one media stream to the content provider for broadcast on the broadcast media channel;

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

h. wherein the web page is rendered in parallel in multiple threads.

[137] Regarding Claims 4[a] to 4[f] the Petition at Pages 34-35 point to the same reasons used for 1[a] to 1[f] hence, the reason is the same.

[138] Regarding Claim 4[g], requires that during the "assembling the at least one media stream comprises inserting at least one MPEG packet including metadata corresponding to the multimedia item," as explain from Ex1001 FIG. 9, the inserting

of MPEG metadata is done in a particular "Program Identifier" by modifying the "PMT" or the "Program Management Table" *Id.* 11:62-57 and *Id.* 12:1-10

[139] Therefore, by no means Avellan provides a way to insert in any "program identifier" or even use MPEG as a transport stream. As the Avellan description is far off from how an MPEG Transport Stream works (See EX2040, 262-263)

[140] Regarding Claim 5 "*The computer-implemented method of claim 4, wherein the at least one media stream is interacted with MPEG two way control messages*." As shown in FIG 9, the metadata payload of an MPEG two way message or an MPEG metadata will include a link to a web-site or an specific "URI". This link will allow the "media stream" to interact with MPEG two way control messages, as it contains a URL that the user's device can decode and passed to the devices' browser. None of these is shown by Avellan.

[141] Hence, at Petition Pg. 36, the reasons for Claim 2 also apply.

[142] Regarding Claim 6 "*The computer-implemented method of claim 4, wherein MPEG messages are sent to a service provider via a unicast address*" Petition, 36 shows that the same reasoning for Claim 3 applies hence the same is true here.

[143] Regarding Claim 13 "A computer-implemented method, comprising:

a. 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;

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

c. rendering a web page by a browser using the content;

d. 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;

e. assembling the at least one media stream using the temporal sequence of screen captures; and

f. providing the at least one media stream to the content provider for broadcast on the broadcast media channel;

g. 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

h. wherein the web page is rendered in parallel in virtual instances or virtual machines.

[144] Petition,37 uses reasons for Claim 13-pre to 13g for 1-pre to 1g. Hence the same reasons apply here.

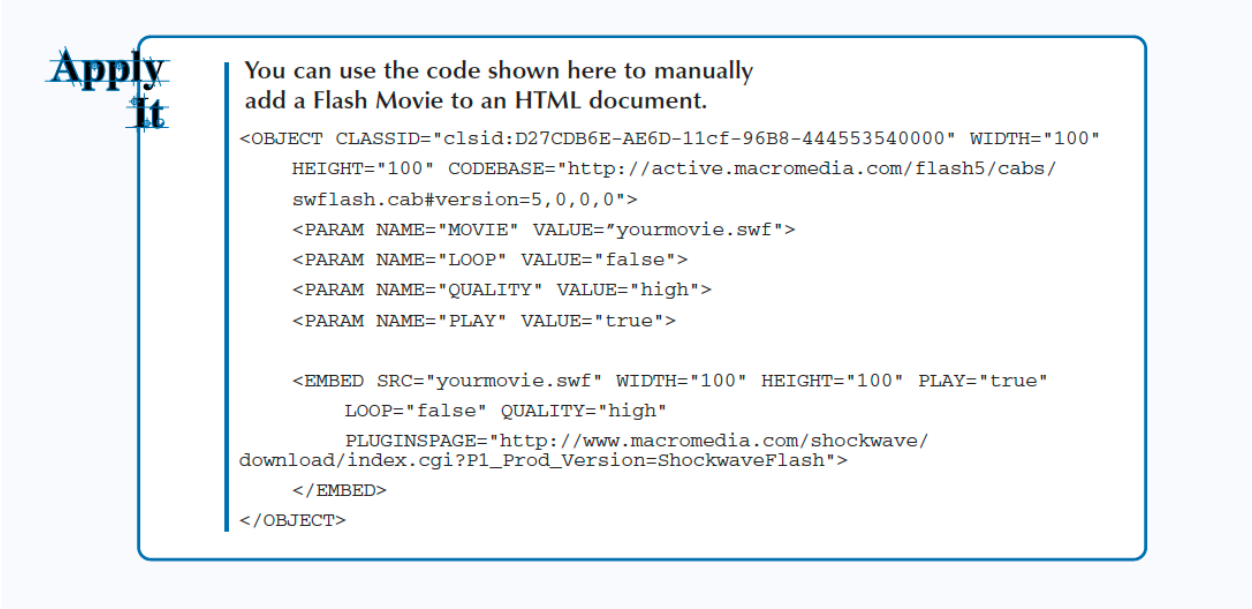
[145] For Element 13h the same reasons apply for 1[h], however Petition, 39 states that:

> 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.*

Petition, 40

[146] As shown, the Ex-1007 ¶137 is mistaken, as the use of a "Java Virtual Machine" is not referenced by Avellan. Instead, Avellan discusses data on a web page that includes, html, java, flash. In general, Java refers to "Java Applets" embedded as html, as well as "Flash" embedded web video players.

[147] For example, a way to insert an "Adobe Flash Video Player" is shown at EX2041 at Pg. 35 shows:



And similarly, an Applet on a web page looks like this,

```
<html….. >
<applet code="Example.class" width="300" height="100">
     Your browser does not support Java applets.
  </applet>
</html>
```

[148] Similarly, as shown in EX2041,35, the example shows the use of html tag: "<OBJECT CLASSID>" and, similarly Java Applets are embedded as part of HTML. Many of these tags need to be supported by the "Browser"

[149] Moreover, a Java Applet is embedded into HTML in a similar way as "Flash objects" to handle for example, the creation of a "user interface" and reflect those changes on a web page. These applets are downloadable applets which are obtained upon request. The reference Ex1006 [0005] confirms the use of Java applets on web pges.:

> for the set top terminal. The menu structure has each menu (e.g., menu instructions, graphics and video) contained in downloadable applets which are sent upon request from the service provider equipment to the set top terminal for display. As such, the navigator functions are actually dis-

Ex 1005, [0005].

[150] Besides Ex.1005, Ex1010 at 11:21-27 shows that the "HTML data object… include resources such as stylesheets, JavaScript files, applets, flash files, images, video, audio…." are part of HTML or a web page.

> ment such as an HTML data object. In either case, multiple application layer protocol request response sequences are preferably executed to retrieve the referenced, embedded, or otherwise included resources such as stylesheets, JavaScript files, applets, flash files, images, video, audio, fonts, and/or other suitable components as would be obtained when rendering a webpage. Transmitting a content request preferably

[151] Hence, a "Virtual Machine" is not suggested for the "Gateway 130" nor the use of the word "Virtual Browser" implies that a "virtualization" environment is in use. Virtual Browser is used, as the application residing on Avellan is a "pseudo-Browser" as Ex-1002,78:60-64 shows.

[152] Hence, Claim 13 is not anticipated by Avellan.

B. Ground 2: Claims 1-6, 13 are not obvious under of Avellan, in view of Ma, and Fain.

[153] Regarding Claim 1[pre] to 1[f], Petition, 40 uses the Ground 1 analysis.

[154] Petitioner focuses starts with Element 1[g] that reads "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,

[155] First, Avellan and Ma are not combinable, instead they both provide opposites, for starters Ma is related to "media streams". HLS, m3u8, DASH, e.g. Ex1004 [0033] and Avellan to "compressed files." Ma further speaks about "live stream encoder 202"

[156] A POSITA would not have been able to combine Ma and Avellan, as Avellan would attempt to retrieve the HTTP live stream or DASH, as web page screen shots and the image being captured and compressed. However, Petition, 41 mistakenly states that "HLS… transmits video via HTTP." This is incorrect, as HTTP is pull-based protocol that requires a user's request first to an index file called manifest or m3u8 (Ex-1004 [0031]) and that is used as a index. The M3U8 is a text file, when a client web-browser can request from a web server.

[157] Hence, a POSITA would encounter that "Additionally, Avellan discloses transmission of live content, including "live streaming television feeds" and "live broadcasting." Ex1003, 8:18-21, and will see that Avellan is "compressing" the

encoded M3U8 feed, creating a degraded user experience from what the original HTTP Stream provides.

[158] Hence POSITA would not combine Ma and Avellan, as Avellan simply will create "compressed image captures" of "adaptive bit rate" streams in HLS, eliminating all bit-rate adaptiveness and creating an issue for video. Without mentioning, how audio would be handled, as well as Web Player integration.

[159] As shown at Ex-1004 [009], a "**media player**" is required to handle M3U8 extraction and integration, as manifest files are being rendered in real-time.

> **[0009]** In another aspect, a method is provided for dynamically extracting advertisement metadata from segments and manifest files being rendered in real-time, to enable targeted advertisement replacement. In one embodiment, a client proxies manifest file requests from a local media player and extracts advertisement metadata from a received manifest before forwarding it to the media player (e.g., extracting comments from an m3u8 manifest). In another embodiment, the client proxies segment requests from the media player and extracts advertisement metadata from the segment before forwarding it to the media player (e.g., extracting proprietary

Ex1004 [0009]

[160] Avellan destroys the efforts gained by HLS and the M3U8 link used by Ma. Ma appears to be properly using HLS, DASH, and other protocols to handle encoding for a particular bitrate, for example. The combination with Avellan creates

two competing technologies a) compressed image from a web page that contains a live stream (e.g. HLS) versus b) using the live stream directly. As the result is obviously inferior, as for example, the gateway server in Avellan uses 1Gbps link, as Avellan suggest, then the Adaptive bit rate for Ma, would return a 4K movie, and if the satellite link is 128Kbps

[161] As shown Ma's uses the manifest and multi-bit-rate feature, as in *Id.* [0044]

> **[0044]** FIG. **2** is a block diagram of a live stream processor **102** for one embodiment of the present invention for implementing a live stream processor **102**. The live stream processor **102** includes multiple components, including a live stream recorder **202** which records source audio/video content, a multi-bitrate transcoder **204** which transcodes source content into different bitrate encodings, a multi-bitrate segmenter **206** which creates segment files from transcoded content, an inline metadata insertion module **210** which inserts metadata into segment files, a segment encryptor **212** which encrypts segments prior to uploading them to a CDN **104**, a manifest generator **208** which creates manifest files to

Ex1004 [0044]

[162] Therefore, under Ground 2, Based on Ground 1 and the analysis of Element 1[g], Ground 2 does not disclose Claim 1.

[163] For Element 1[h], the book by Fain is cited.

[164] However, as explained earlier there is no known Java browser, except for Android or Davlik VM (See EX2060, 28) includes some Java APIs including **android.webkit.webclient** modules (EX2043, 1) similar to JavaFX that embeds a

browser into an "App", which would require undue experimentation to implement "Avellan" to use this technology, and creating modules for Radio Interface Layer (RIL) to work with a satellite link, plus any changes made to the User's computers 104 and 106 at Ex1004 FIG 1.

[165] Hence, the citation that Avellan uses Java ***is not accurate***, as the HTTP or web page could have Java, Flash, HTML and other objects.

> 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

Petition, 43.

[166] The citation from Avellan clearly shows, that "At step 208, the server 134 creates a compressed video format of the web page…. The compressed video format may include all of the page's graphical information and data…. The data may include Java, html, flash" Ex1003 36-41

> At step **208**, the server **134** creates a compressed video format of the web page. The compressed video format may include all of the page's graphical information and data from different sources in a single video signal representing the image of the web page. The data may include Java, html, flash, or other programs known to those of skill in the art. Also

[167] See Ground 1 for Claim 13[h] where I discuss how flash, Java, and other objects are embedded into the HTML.

[168] Regarding Claim 2 language "*The method of claim 1, "The method according to claim 1, wherein the at least one media stream is interacted with MPEG two way control messages*"

[169] Although Ma, at Petition, 45 discloses the use of MPEG Transport Streams, the way the M3U8 manifests insert ads are using the EXT-X-MEDIA:ID3 tag, based on the RFC8216[6]

a. For example, to insert **for ID3 tag:**

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-MEDIA-SEQUENCE:0
#EXT-X-TARGETDURATION:10
#EXTINF:10.0,
#EXT-X-MEDIA:ID3="TIT2=Sample Song; TPE1=Sample Artist; TALB=Sample Album"
media_0.ts
#EXTINF:10.0,
media_1.ts
#EXTINF:10.0,
media_2.ts
#EXT-X-ENDLIST
```

[6] https://datatracker.ietf.org/doc/html/rfc8216 (EX2056)

b. For Ad insertion:

ID3 TAG in M3U8 Index Files

#EXTM3U

#EXT-X-VERSION:3

#EXT-X-MEDIA-SEQUENCE:0

#EXT-X-TARGETDURATION:10

Main content segment

#EXTINF:10.0,

media_0.ts

#EXTINF:10.0,

media_1.ts

 # Ad insertion (discontinuity indicates a break from main content)

#EXT-X-DISCONTINUITY

#EXTINF:15.0,

ad_segment_0.ts

#EXTINF:15.0,

ad_segment_1.ts

Return to main content

#EXT-X-DISCONTINUITY

#EXTINF:10.0,

media_2.ts

#EXTINF:10.0,

media_3.ts

#EXT-X-ENDLIST

[170] Hence, if an MPEG Packet is inserted with a "Program Identifier" (PID) and the "Program Management Table" (PMT) is not shown by Ma, instead Ma automates the generation of the index file or M3U8 or text files, as Ex1004 shows FIG.2 Element 218 is "Manifest Metadata Insertion" which means is acting on the M3U8 index file or ***text file***. Inserting an MPEG Transport Stream packet requires data structures and is a ***binary*** operation to handle Elementary Streams and 188-byte packet format.

[171] Again, Avellan will capture from a media player the DASH, or HTTP Live streaming and such combination will be in decrement of using the live steam directly, hence a POSITA will not have a motivation to combine as the result will not improve the system and video quality.

[172] Once again, such combination with Ma will turn into ***"image screen captures***" that will be turned into "compressed video files" and "tag files" from the embedded player that Ma also suggest the use of a local Media Player Ex1004, [0009].

[173] Regarding Claim 3 language "The method according to claim 1, wherein MPEG messages are sent to a service provider via a unicast address.

[174] POSITA will not have a motivation to combine as the result will not improve the system and video quality as Ma works in a live stream and Avellan is not required.

[175] Regarding Claim 4 language: A computer-implemented method, comprising:

a. 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;

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

c. rendering a web page by a browser using the content;

d. 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;

e. assembling the at least one media stream using the temporal sequence of screen captures; and

f. providing the at least one media stream to the content provider for broadcast on the broadcast media channel;

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

h. wherein the web page is rendered in parallel in multiple threads.

[176] For Claim 4, 4[pre]-4[f] are the same reasons as for Avellan, Ground 1, the same analysis returns no anticipation by Avellan.

[177] When at least 4[g], Petition, 47, Avellan, Fain and Ma, Ex1007¶160, as Ma does not show Metadata Insertion into the MPEG TS, then as the analysis does not include Fain, and Ma has been established for Claims 3 and 4 that is not a combination that a POSITA will find counterproductive.

[178] Claims 4,5, 6 use the same reasoning for 1[h], 3, and 3, that have been demonstrated not anticipated.

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

[180] As indicated by the ‘002 patent, MPEG packets can be inserted at anytime of any type, including MPEG two-way messages as shown at Ex1001, 11:35-40.

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

[182] Regarding Claim 13 “A computer-implemented method, comprising:

a. 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;

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

c. rendering a web page by a browser using the content;

d. 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;

e. assembling the at least one media stream using the temporal sequence of screen captures; and

f. providing the at least one media stream to the content provider for broadcast on the broadcast media channel;

g. 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

h. wherein the web page is rendered in parallel in virtual instances or virtual machines.

[183] Element 13[pre] to 13[f] are based on Ground 1, that is not anticipated.

[184] Element 13[g] is based on 1[g]

[185] Element 13[h] is not met, as Avellan does not disclose that is a Java-based platform, nor that it can be rendered in parallel, as the Web-to-HD Video Protocol is not defined and cannot be determined if it can be instantiated in a "virtual machine"

[186] Hence Claim 13 is not anticipated by Ground 2

C. Ground 3: Claim 7-9 and 13 are not obvious under Wannamaker

[187] Claim 7 recites: [Pre] A system, comprising:

a. 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;

b. a caching server communicatively coupled to the multicast server, and configured to

c. receive and store the parameters,

d. 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,

e. assemble the temporal sequence of screen captures, and

f. 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

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

[188] Wannamaker uses a **"microbrowser**" as referenced in the Petition, 18 and Ex1006, [0267]-[0261] at the client's computer.

[189] First, Claim 7[a] at "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;" and which Petition, 50 points to "The data carousel server is the multicast server and includes MMS components for the data transmission"

[190] Hence, the "multicast server is the "data carousel server" and MMS components appears to be the "configured to generate parameters at least one… configured to generate MPEG Outputs"

[191] As shown by Ex1006 at [0168} the "Mediacast (Data Carousel) Server," Medacast is a trademark name as shown by [0169]. Mediacast servers according to E1006[00183] employ the "Mediacast Signaling Protocol" (MSP) Objects.

[192] The MSP Objects are described at *Id.* [0183] states that "MSP Namespaces, carousels, and transports are configured via the XML-configuration file"

[193] Hence, there MSP Objects that are cited in Petition, 51 are not "video content, image content, or audio content"

[194] The MMS Server cited in Petition, 50 says:

> 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

[195] However, an MMS server contains "out-band servers (MMS_OM1000) and the Downstream Router" Ex-1006 [0193]., and require an "MPEG multiplexer

(MUX) Card", controlled by the AVTS" Id. [0195], where the AVTS provides the following: "play MPEG I-frames, play MPEG B-frames"

[196] The MMS includes AVTS and AVTS uses an MPEG Card identified as "MPEG Card (LMC) " as in *Id.* [0209]-[0210].

[197] The LMC card is a PCI board as described by *Id.* [0210]

> **[0210]** The information platform uses the information platform drop and insert LMC. An information platform PCI adapter based on the PowerPC CPU performs the MPEG multiplexing. The information platform drop and insert LMC is capable of accepting one input and producing four output streams. Each output stream is generated independently so users on different cable system segments can be isolated, thus expanding the bandwidth available per segment. Drivers for LMC are available for SPARC Solaris. The LMC board is a PCI-based adapter card that is MPEG-2 transport stream compliant and can concurrently transmit and receive MPEG-2 transport streams. The board uses the PowerPC 403GCX as its on-board processor. The primary on-board memory is high-speed SRAM with a 128 MB-per-second burst rate. The on-board processor off-loads scheduling of user data requests from the host system.
>
> **[0211]** The LMC performance can be summarized as follows:
>
> **[0212]** 100% usage of four concurrent MPEG-2 streams at the 64 QAM rate of 26.97 Mbps or the 256 QAM rate of 38.81 Mbps.
>
> **[0213]** The four output streams can be combined using the information platform interface translation PCI board to form single DVB-ASI stream with up to 107.88146 Mbps payload data rate (DVB-ASI transport rate is 270 Mbps)

Ex1006 [0210] to [0213].

[198] As shown above, the OM1000 with is an out-band server as shown at *Id.* [0227] to [0228].

[0227] MMS_OM1000

[0228] This server communicates with specialized hardware to deliver out-of-band data, such as updates to the set-top box internal data store, to the set-top box. Any request that can be sent via the in-band data channel may be sent out-of-band. Generally, the MMS_OM1000 is used to send OOB directories and files, subjecting all data to rate-throttling to prevent buffer overflow in the set-top boxes.

[0229] The data may be:

[0230] Broadcast to all set-top boxes

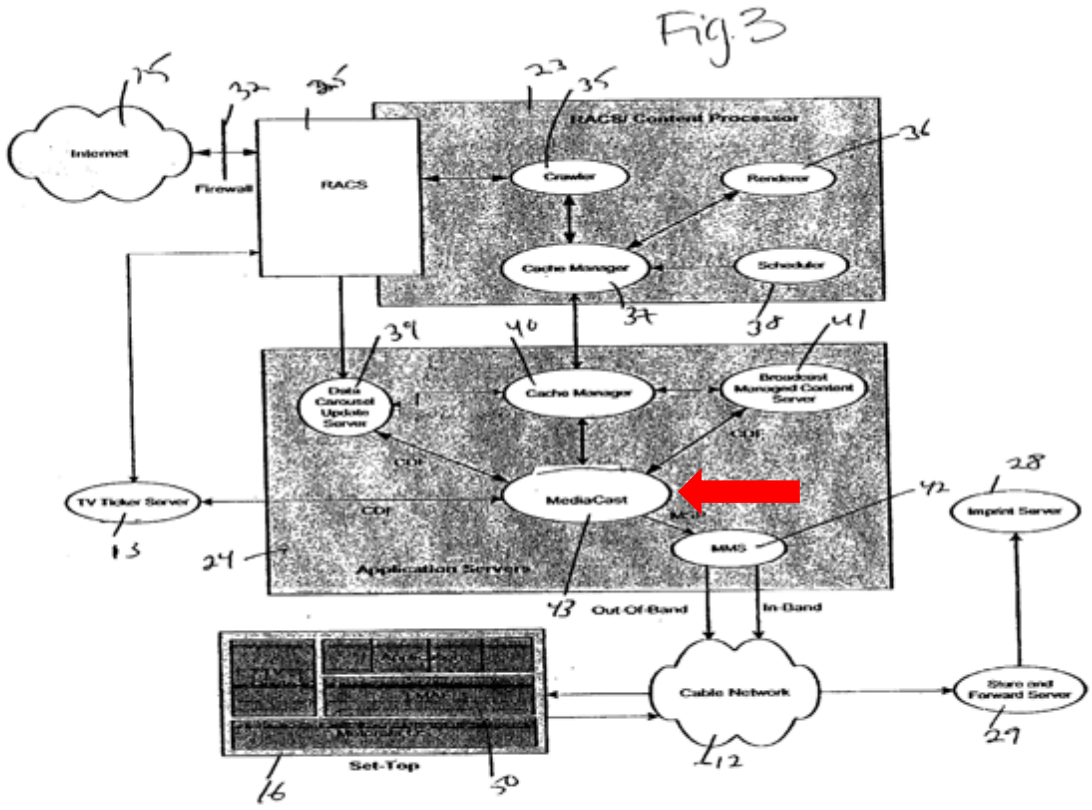
[0231] Multicast to a group of set-top boxes

[0232] Single-cast or "addressed" to a single set-top box

Ex1006 [0027] to [0232]

[199] Petition fails to point to the MMS_OM1000 as the out-of-band encoder there is the only device capable of creating a multicast to a "group" of set top boxes.

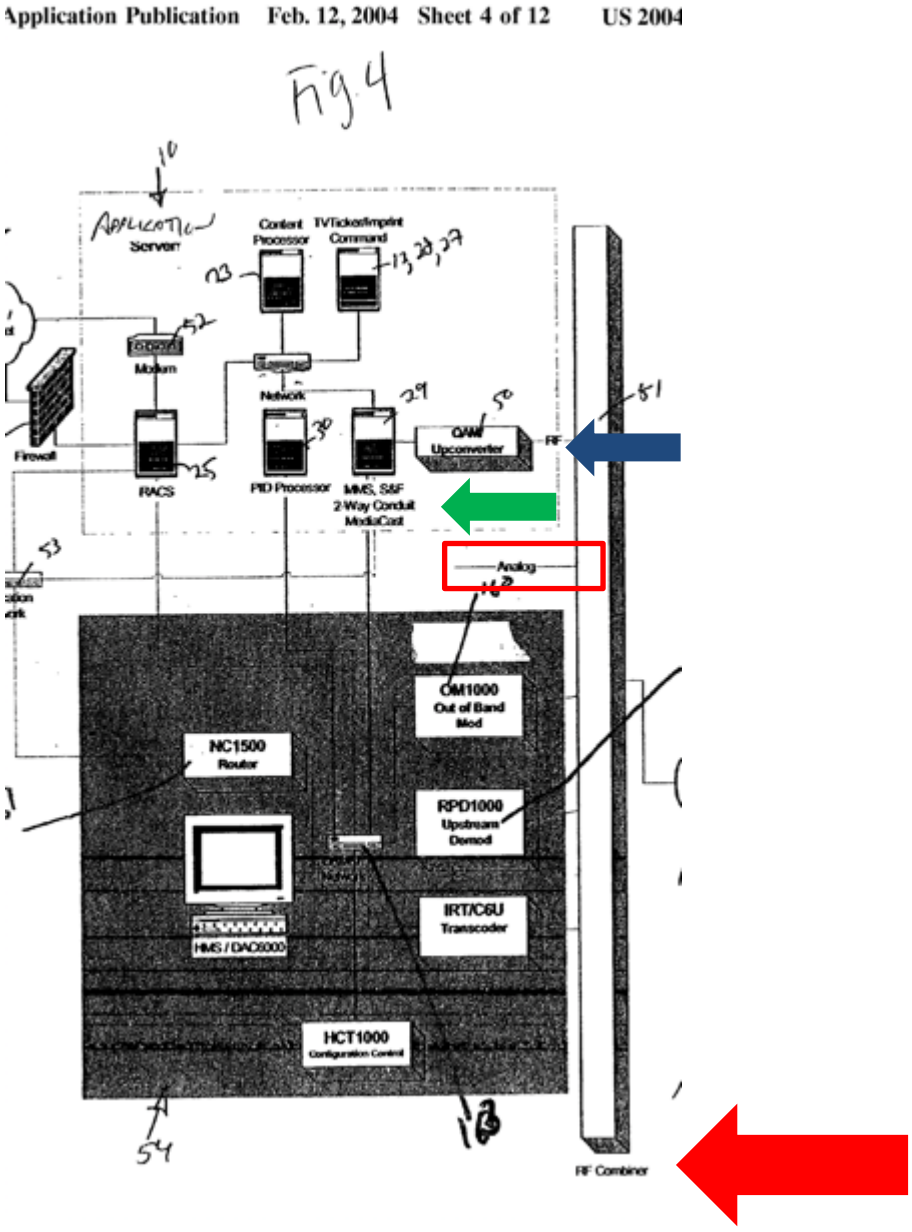
[200] Ex1006 at FIG 3 shows how the "Mediacast Sever" interacts with MMS using MSP packets (in red).



Ex1006 FIG 3.

[201] What this means is that the LMC card works with 64 QAM rate of 27.9Mbpx or 256 QAM at a rate of 30Mbps. The use of modulation is because the systemis now CATV compatible for Cable TV. In other words, the MMS works with the out-of-band traffic.

[202] As shown in FIG 4,



Section of Ex1006 at FIG 4

[203] Observe that the "RF Combiner is output for all elements" (Red Arrow) what this means is that, the QAM Upconverter (blue arrow) that the MMS (green arrow) is connected to a "QAM upconverter" with the number 50 next to it.

[204] Therefore, the output of the MMS system and hence from Wannamaker is not a Multicast, but instead, an RF stream for an analog system as in CATV.

[205] This is confirmed by *Id.* [0123] to [0130], as in Element 50 connected to the MMS is the QAM Modulator/RF Upconverter, the OM2000 160 is the out-of-band modulator

> **[0123]** Downlink
>
> **[0124]** **FIG. 6** shows the data path through the down link and out to the set-top box.
>
> **[0125]** The down link site must have the following (see **FIG. 4**):
>
> **[0126]** QAM Modulator/RF Upconverter **50**
>
> **[0127]** OM2000 Out-of-Band Modulator **160**
>
> **[0128]** NC1500 Network Controller **161**
>
> **[0129]** RPDx000 (Reverse Path Demodulators) **162**
>
> **[0130]** OAM&P Network **163**

[206] The Motorola OM1000 module used by the MMS block is provided herein:



OM1000
MPEG-2/Digital Out-Of-Band Modulator

MOTOROLA
intelligence everywhere

Out-of-band modulator used to transmit MPEG-2 data streams to cable terminals

The Motorola out-of-band multiplexer/modulator, OM-1000, provides an out-of-band data stream to the digital settop terminals. It is primarily used as a signaling channel but can also provide for transmission of program guide information, code downloads, InfraRed codes, as well as application information. It is also utilized as the downstream path in an interactive system.

The input to the Motorola OM-1000 can be configured with a mix of both Ethernet and serial inputs. A Quadrature Phase Shift Key Modulator (QPSK) modulates the data on a 1.5 MHz wide carrier. The Motorola OM-1000 is agile across the 71 - 129 MHz band. The transmitted data packets are composed of data that can be multiplexed either in the Motorola OM-1000 or externally. The Motorola OM-1000 uses null packets as necessary to perform rate aggregation between its inputs and the 2.048 Mbps output. Data integrity is protected by Forward Error Correction encoding and interleaving.

HIGHLIGHTS INCLUDE:

- Transmits MPEG-2 data streams to cable terminals individually or on a broadcast basis
- Data streams consist of authorization information, electronic program guide information, code downloads and interactive traffic
- QPSK modulates data stream in a 1.5 MHz carrier to both IF and RF output ports
- Implements FEC code for cable transmission
- Generates and sends emergency alert packets to cable terminals
- Multiple data accepting ports including RS232, RS530, and Ethernet
- Frequency agile RF output with variable power control
- Industry standard 10 Base-T Ethernet connectivity for status monitoring and network element control
- Supports broadcast messaging
- Supports interactive applications

Ex2059 Pg. 1 of OM1000 [7],[8]

[7] https://www.scribd.com/document/361305809/OM1000

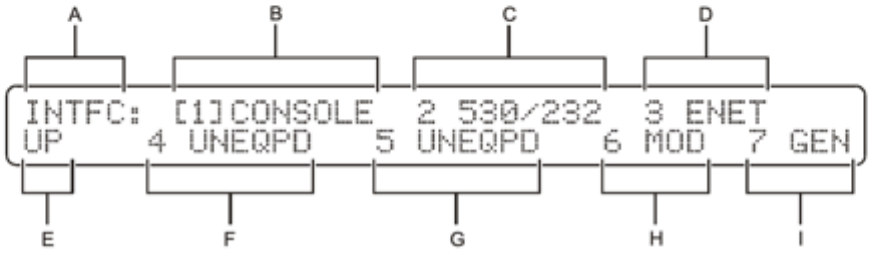
[8] https://www.manualslib.com/manual/1727666/Motorola-Om-1000.html , https://www.manualslib.com/products/Motorola-Om-1000-10772837.html

NTFC Menu—Modifying the Physical Port Default Settings

To modify the physical port default settings:

1 Select **INTFC** on the MAIN menu and then press **ENT** to display the INTFC menu illustrated in Figure 4-4:

Figure 4-4
INTFC menu



2 Press ↑ or ↓ to select the port to configure and then press **ENT**.

3 Use Table 4-4 to set up ports 2, 4, 5, and 6. The table describes the configurable menu items for these ports.

4 After configuring the menu items, select **UP** and then press **ENT** to display the MAIN menu.

Table 4-4 describes the fields and available options:

Table 4-4
Configuring the physical ports

Key	Field	Description
A	INTFC:	Identifies the menu.
B	1 CONSOLE	Displays the internal console settings for PORT 1. The console port is accessible from the rear panel. It provides asynchronous communications at 9600 baud and cannot be reconfigured at the front panel. It is used only for maintenance.
C	2 530/232	Configures the operating parameters for PORT 2. This port is configured for asynchronous communications, and the data rate is configured from 300 bps through 19.2 Kbps.
D	3 ENET	Displays the operating parameters for PORT 3. A display-only field displays the MAC address; a selection for the Time-To-Live (TTL) setting, a selection for enabling or disabling IGMP, and a menu selection for the multicast MAC receive address assigned to the OM 1000 at the factory.
E	UP	Displays the MAIN menu.
F	4	Configures the operating parameters for optional PORT 4. Configure the data and baud options as described for PORT 2, if present. Select async or sync (internal clock, external clock, or slv on rx clock). The INTFC menu displays UNEQPD if the communications board is not present and configured. Refer to the `operate.txt` file in Appendix C, "Initialization Information."

EX2044 at Pg. 37/97 of the installation and operation manual of OM1000.

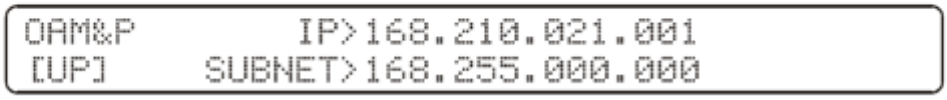
Using the Front-Panel Interface

The OM 1000 front-panel interface includes an LCD and a set of cursor control keys that enable you to set and change operating parameters. The LCD provides a series of menus showing specific OM 1000 parameter settings. The menus provide:

- Display-only information that you cannot select or change
- Parameter settings that you can select and modify
- Navigational aids that enable access to other menus

Figure 4-1 illustrates a typical display. The menu name, OAM&P, is display-only; the SUBNET and IP options are changeable parameters.

Figure 4-1
Sample OM 1000 configuration menu



To branch to other menus, use the UP and MORE (not shown) options as navigation aids.

EX2044 Pg. 29/97 of the installation and operation manual of OM 1000[9].

[207] As shown above, the OM 1000 is manually configured thru the "Front panel interface" as shown above. **This shows an obvious technology mismatch, from hardware with a method patents.**

[208] The OM1000 can only "receive a multicast" and cannot send one, plus the "Ethernet interface" can onby be used for management purposes.

[9] Source https://infiniumnet.com/PublicDocs/motorola%20om1000%20manual.pdf The same manaul is found at "ManualLib.com" See: https://www.manualslib.com/manual/1727666/Motorola-Om-1000.html?page=29#manual

The OM 1000:

- Receives Ethernet User Datagram Protocol (UDP) packets from a controlling processor
- Multiplexes downstream data from multiple sources
- Provides forward error correction (FEC) encoding of the bitstream, as well as interleaving and randomization
- Provides downstream QPSK-modulated output within the 71 to 129 MHz range
- Provides industry-standard Ethernet 10Base-T connectivity for operation, administration, maintenance, and provisioning (OAM&P)
- Performs periodic insertion of internally stored messages
- Controls and provides status/alarm reporting using Simple Network Management Protocol (SNMP)

Id. Pg. 10/97

[209] The '002 Patent teaches that the "Multicast Server" consists of:

a. The Multicast Server or Units "can pull or obtain information from the caching unit using via TCP/IP or UDP, using protocols such as HTTP (Hypertext Transport Protocol) or RTSP (Real-time Transport Protocol" Ex1001,5:4-7

b. The Multicast Server provides content to a Multiplexer or MSO (*Id.* Element 230 FIG 2).

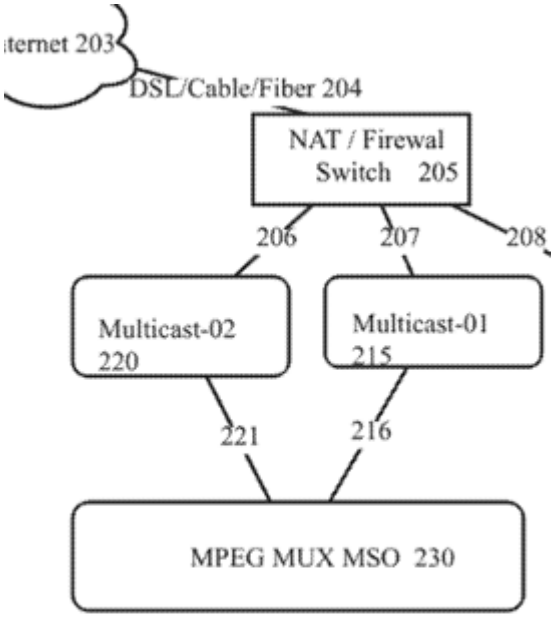


FIG. 2

c. The Multicast Server or Units broadcast using multicast addresses, "In general, the 239.*.*.* network address maybe routed to those interfaces" *Id.* 5:42-44

d. The Multicast Server can also receive multicast packets (*Id*. 9:55-56), and handle other protocols such as HTTP (*Id.* 9:41-45)

e. The Multicast Server uses MPEG Transport Streams for multicasting and other broadcasting protocols (*Id.* 10:65-47) that are compatible with the "Caching server" specifications.

[210] As a consequence, The MMS with MMS_OM1000 fails to be "a multicast server" as the MMS_OM1000 cannot even create a "Multicast Address" stream, only receives it.

[211] As shown in the Petition, 53, also points to the "Mediacast Server" however, the "MediaCast" server uses the MMS server, or MMS_OM1000 device to handle MPEG content.

[212] For Element 1[b], "a caching server communicatively coupled to the multicast server, and configured to:"

[213] Petition, 54 appears to point to the "CP or Content Processor" to be the "Caching server and configured to;"

[214] Element 7[c], "receive and store the parameters," as shown

a. First Petitioner at Pg. 55 points to the "Cache Managers" (37 and 40 at *Id.* FIG. 3, and the "CP-Content Processor" from FIG. 3.

b. Second, although the "CP-Content Processor" uses a "Cache Manager" 37 and provides content to the TV Navigator to the "micro browser " that in Wannamaker was implemented in a Motorola DCT 2000 at *Id.*[0332]

c. Third, Petition, 57 states that "the crawler of the content processor…" and concludes that the "content processor inherently receives parameters generated"

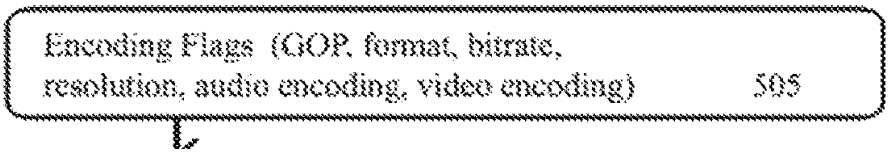
d. Fourth, Petition, 56 points to "Therefore, the cache managers of the content processor ("cache server") receive the message ("parameter") from the application server (multicast server). Ex-1007, ¶173

[215] None of the parameters provided by the petition are correct, as they are unrelated to the process of encoding and transcoding carried by the Multicast Server and Caching unit, instead the '002 patent teaches that:

available from the unit. FIG. 5 illustrates an example method
that can be implemented by the Mediaplug, and in particular,
35 by the "Caching" unit. At step **505**, encoding flags or input
parameters can be received which may include GOP (Group
of Pictures) which is used by certain video encoders, reso-
lution (e.g. 640×480) in pixels, bit rate in Kilobits per
second, any encoding type of video (e.g. H.264), encoding
40 type of audio (e.g. AC-3), duration of the video feed, etc. In

[216] Encoding flags or input parameters may include GOP (Group of Pictures), which is used by certain video encoders, resolution (e.g. 640x480) in pixel, bit rate in Kilobits per second, any encoding type of video (e.g. H.264), encoding type of

audip (e.g. AC-3) Ex1001 6:34-40, also *Id.* at FIG. 5 Element 505 reflect this teachings.



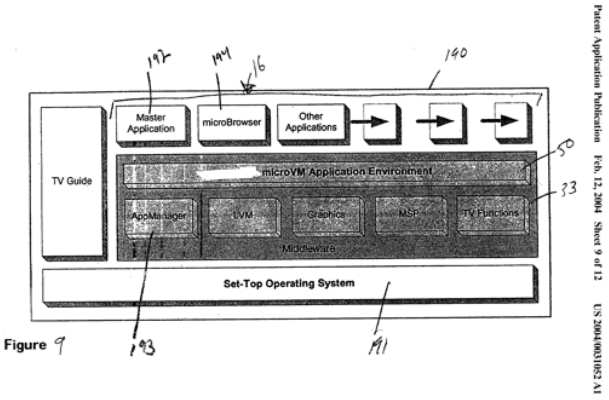
[217] These encoding flags or parameters are used to generate the "stream" for multicasting via MPEG outputs, and are also stored for future use.

[218] Hence, the Petition and Mr. Lipoff declaration dose not disclose any type of encoding parameter. Instead points to various "parameters" that are not needed for Element 7[c].

[219] Encoding is handled by the OM1000 device not by the "Content Processor" or the "Cache Managers," hence Ex1006 does not suggest or implies that those parameters are going to be stored in the "Content Processor" or in the "Cache Manager" as the petition cites.

[220] However, the Petition and Mr. Lipoff simply points to many types of parameters expecting to guess the right one.

[221] As shown herein, FIG. 9 depicts, the architecture where the "microbrowser" 144 is located as part of the Set Top Box and the "TV Navigator"



[222] Therefore, for Element 7[d] and 7[e] are combined in the Petition, 57

[223] First, element 7d] recites "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." and 7[e] recites "assemble the temporal sequence of screen captures, and"

[224] Petition, 57, in reality the TV Navigator is the ultimate recipient of the content that the CP-Content Processor creates for the set top box.

[225] Ex1006 [00259] shows that an HTML page is "Rendered" but into a CPML or MML page with FMB files. Mr.. Lipoff did not know what "FBM", "Outout CPML" or "MML" pages were (EX2054, 42:4-12). According to *Id.* [0660], CPML is Information Platform Markup Language (CP is the acronym used for Content

Processor, hence a Content Processor Markup Language), as well as MML, another markup language that is proprietary and referenced by Ex1006 as well as "FMB" which is an image file format (*Id.* [0259]).

[226] Hence, the Rendering process disclosed by Ex1006 does not generate any sequence of images, instead generates or translates complex HTML to MML or CPML markup languages such that a "microbrowser" can handle a simpler set of "tags" , and converts animated gifs and files to FMB image format. A POSITA will understand that this process is called "Transcoding"

[227] In fact, *Id.* [273] further confirms that Ex1006 has a microbrowser that processes MML (converted from HTML) and MPEG image data, as well as MML permits the user navigate hypertext links.

> **[0273]** 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.

Ex1006 [0273]

[228] Even in more detail, *Id.* [0489] and [0490] explain that there is no rendering done into "screen captures" from a web pages, even less a rendering related to a sequence of images. Instead the HTML is converted to CPML or MML such that the

microbrowsers visualizes "content has been authored in HTML" as "server's software handles retrieving and ***transcoding*** HTML content.." (meaning the Content Processor)

[229] Therefore, transcoding means that there are no screen captures, instead it means "conversion from one format to the other", e.g. HTML to MML/CPML as explained before. Obviously, a microbrowser was unable to process heavy HTML but was able to handle MML and CPML as disclosed by *Id.* [0490] "The sever software handles **retrieving and transcoding** HTML content on a set schedule" (emphasis added)

> [**0489**] Microbrowser
>
> [**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. The server software handles retrieving and transcoding HTML content on a set schedule. This transcoded content is then inserted into the multicast file system for use by the microbrowser. 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. When users exit an application, they are returned to the Web page from which the application was initiated. Because much of the content can be pre-cached, managed content can avoid the long latencies associated with the Internet and produce response times of typically less than one second. **FIG. 11** shows a typical microbrowser screen.

[230] Hence, in Wannamaker "Rendering" is done by the Content Processor and is nothing but "Server Transcoding" Id, [258-259] that "to produce output image files (FMB) and output **CPML or MML**" where **CPML** stands for "Information Platform Markup Language" *Id.* [0660]

[231] This is like what WAP systems did in the past, before Smartphones were able to handle HTML directly. Companies like OpenWAVE and Blackberry transcoded the web (EX2043, 2)

[232] I worked at Motorola Mobility in early 2000s and was in charge of several projects including the WAP-PUSH for Android. In my personal experience, as of 2008-2009 when Android arrived is when transcoded systems were no longer required.

[233] Serve Transcoding definitions is clear by Wannamaker but also at EX2043 (War of Mobile Browsers authored by myself at IEEE Pervasive Computing Magazine)[10]

> **Server transcoding.** In this approach, a mobile client requesting a URL is sent to a proxy server on the Internet, usually residing at the carrier premises. The proxy server, on behalf of the client, makes the request to the encoded server on the URL. HTML content retrieved is converted into a special or proprietary format and rendered by the requesting mobile browser. In general, these rendering engines are proprietary and reside on the server. The mobile

[10] E. A. Hernandez, "War of the Mobile Browsers," in IEEE Pervasive Computing, vol. 8, no. 1, pp. 82-85, Jan.-March 2009, doi: 10.1109/MPRV.2009.21.

> browser might also include additional compression to support low-bandwidth scenarios. (**emphasis added**)
>
> Ex2043 at Pg. 1

[234] Clearly, the Petition points to Wannamaker' "Content Processor" server-based transcoding and by no means this process is related to "***Screen Captures***" from a web page, as Wannamaker teaches that "links" are sent to the Set Top Box, hence Markup Language information is sent in the MPEG Packets as well as images.

[235] Therefore, Element 7[d] and 7[e] are not disclosed by Wannamaker.

[236] Regarding, Element 7[f] "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 despite the Petition analysis at Pg. 59 and 62, simply the CATV and RF nature (analog) of the outputs in Wannamaker (See Fig. 3) and QAM upconverter teaches away from using any "Multicast Stream," although MPEG encoded video is cited in Ex1006 at [0272]. FMB files are image format files, MMS is uses the OM1000 that can't multicast anything except receiving a multicast. Even if Petitioner would argue that MPEG I-frames are part of the web page images only and no markup language is used. A) there is no evidence how Wannamaker handles audio and video as the result of a web page "rendering" in fact, Petition, 59 is silent about "audio." The claim states" with MPEG **encoded**

video and audio..". Hence Element 7f, lacks the multicast stream, among other deficiencies as discussed herein.

[237] Hence, Broadcasting in Wannamaker refers to the last step when the MMS uses a QAM upconverter or modulator to transfer all streams to the set-top-boxes.

[238] Finally, Element 7[g] that consists on " a monitoring system configured to maintain generation of video and audio files, even in failure cases." Monitoring refers to the system of "generation of video and audio files," none of this can be disclosed by Wannamaker as the use of OMI or "Open Management Interface" although refers to "Redundancy" via fail over servers as shown by Ex1007 at [0101] to [0103].

[239] Additionally, the '002 Patent defines the term "Mediamplify Parallel Encoder" Ex1001 at 9:10-11, *Id.* FIG 7 Element 705.

[240] The monitoring presented by Element 7[g] refers to maintaining the generation of video and audio files for the Multicast Stream. Multicast Stream is not present in Wannamaker hence cannot be disclosed any monitoring. Besides, Wannamaker does not disclose the use of a "software features" that suffer failures, ***only hardware*** elements; Instead, all monitoring referenced by Petition, 63 is related to OMI, including references from Wannamaker such as "server failures…",

"redundant network components….", and "… named critical hardware… " when the monitoring used for Element 7[g] relates to software failures.

[241] The '002 Patent cites the use of tools such as "**monit**" *Id.* 10:19 that is tied to the processes executing components such as the "streamer" *Id.* 10:20 or Element 850 at FIG 8.

[242] Hence, monitoring is related to Elements 7[a] to 7[f] that are part of the "Mediamplify Parallel Encoder" methods as taught by the '002 patent, and maintining processes such as a "streamer," checks if the "Caching server" is "alive," "Checks if the multicast stream has vanished" *Id.* 10:34-35, among other monitoring processes that generate actions in events of failure.

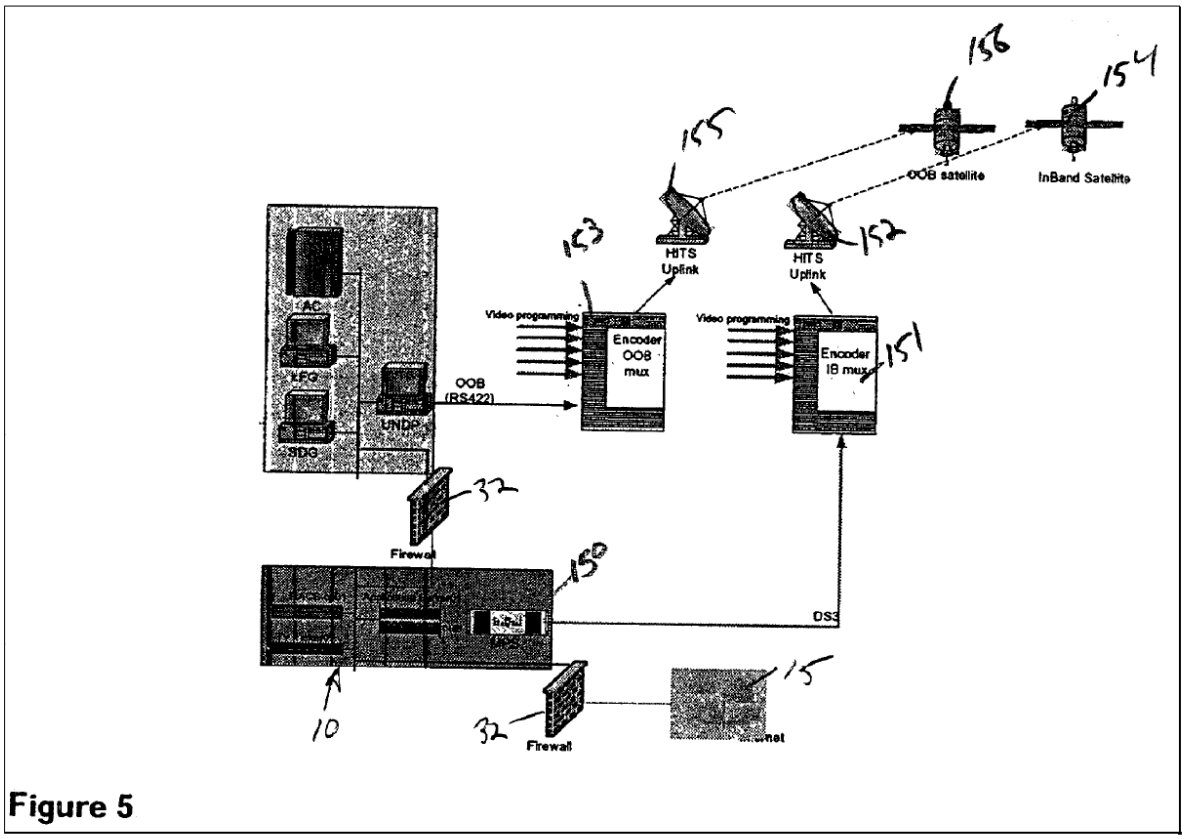
[243] For these reasons, Ground 3 for Claim 7.

[244] Regarding Element 8[a], "The system according to claim 7, further comprising multiple servers for load balancing and fault-tolerance," as nor the Multicast Unit is present and nor the CP-Content Processor. The Petition, 64 points to the same examples used for Element 7[g]

[245] Clearly, 1 or more Multicast Servers can work in conjunction with 1 or more Caching units, depending on the number of channels and availability. In contrast, given the hardware dependency to MPEG encoders (E.g. LMC Cards), is unclear

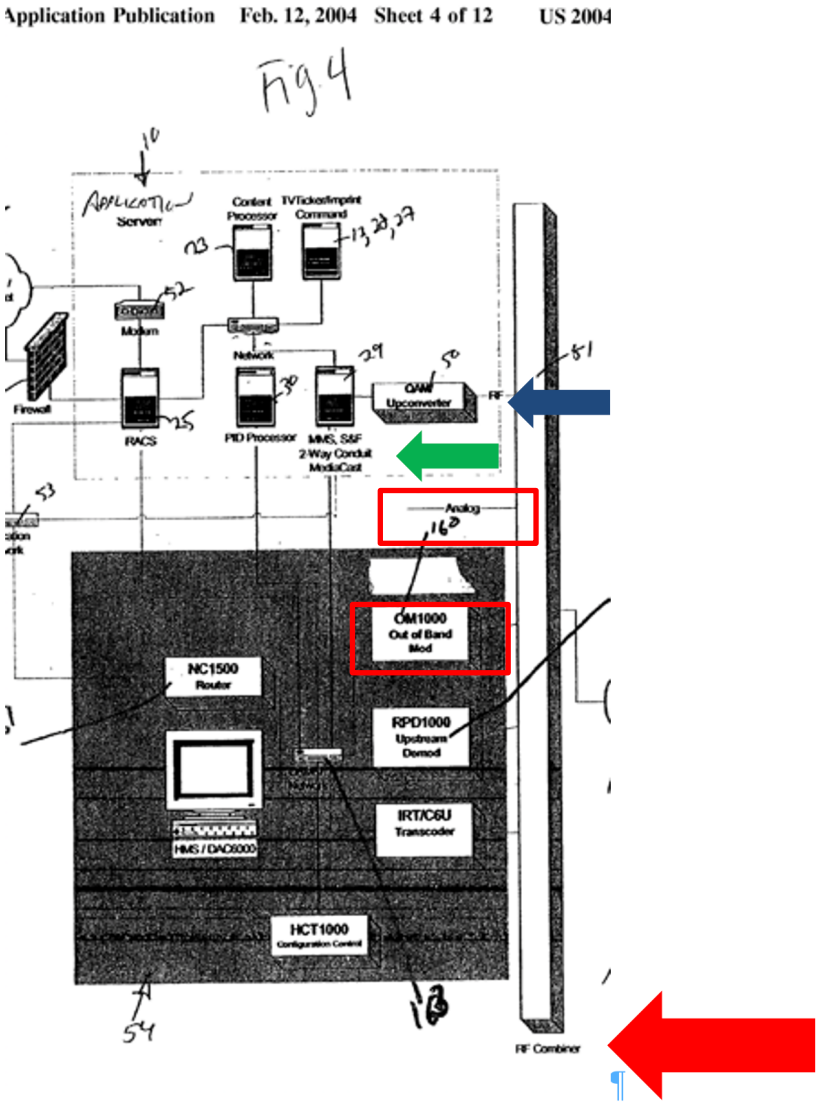
whether more servers will be able to provide "Load-balancing" and "Fault-tolerance."

[246] Regarding Element 9[a] recites "The system according to claim 7, wherein the multicast server is selected to communicate a multicast stream available for a broadband network." As shown by the Petition, 65, Figure 5 shows that Wannamaker needs many more pieces of "hardware" to reach a broadband network.



Ex1007, Figure 5

[247] As expressed earlier, given that Wannamaker MMS's "Green Arrow" server provides an output to the QAM Upconverter or modulator in "Blue Arrow".



EX1006 at FIG 4

[248] Wannamaker cannot be simply exposed to a "broadband network" without major modifications and undue experimentation. As shown, it is the "Application Server" the one that shows how to perform a "satellite link" with encoder 151, as the Petition says at Pg. 65, not the MMS server.

[249] Therefore, Element 9[a] is not anticipated by Wannamaker.

[250] Regarding Element 11[a] that recites "The system according to claim 7, wherein the system generates audio MPEG files and add metadata in an MPEG frame."

> 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

Petition, 66

[251] As Claim 7 is not met, Claim 11[a] is not met also.

[252] As shown in the Petition, 66, A specialized hardware card will have to added to the system to "add metadata in in an MPEG Frame"

[253] As shown, then there is no evidence that shows that "Audio MPEG files" and are added into the MPEG frame. Instead, the '002 Patent teaches that "set 932, an audio PID may include *metadata PID of type DCIIText*, which is added to the stream" Ix1001 12:1-2 and 12:3-6 including FIG. 8 and FIG. 9.

[254] As at least **two PIDs** are involved, additional changes will be required in Wannamaker to the module called "PID" Processor" will also be needed:

> [**0083**] PID processor **30**—The PID processor combines messages on two different PIDs to allow interactive services to be enabled on the DCT2000.

[255] Additionally, the Petition, 67 points to the "transcoded MML" to be the metadata.

[256] For these reasons, Claim 11 is not disclosed by Ground 4

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

[258] As Claim 12 depends on Claim 7, and Claim 7 is valid, then Claim 12 is also valid.

D. Ground 4 –Claims 10 is not obvious under Wannamaker in view of Avellan

[259] Regarding Claim 10 states that "The system according to claim 7, wherein the video content is encoded in a video streaming format so as to generate a unicast stream",

[260] . The combination of Wannamaker and Avellan is not possible, as Wannamaker uses hardware for example the "MMS OC1000" and the microbrowser's CPML/MML and FBM which are "Server Transcoding"

requirements enter into conflicts with the “compressed images and video files” used by Avellan.

[261] Therefore, such combination will fail, as Petition, 72 states “to the user computer 104, where it may be immediately displayed (once decompressed at the computer 104).”)”

[262] However, the “Web to HD Video Protocol” will have be understood with clarity to determine how MPEG encoders in the MediaCAST, MMS, and others are handled.

[263] As Wannamaker “transcoding” of HTML into CPML/MML for the “microbrowser” will enter into a conflict with, video compressed web pages done by Avellan. As two competing formats, then such combination will require undue experimentation and low probability of success

[264] For these reasons Claim 10 is not anticipated by Ground 4.

X. CONCLUSION

[265] Grounds 1 to 9 fail to show that any of the claims of the ‘002 patent is invalid user U.S.C. 102 and 103 grounds.

[266] I may modify or supplement my opinions, if necessary, based on further review and analysis of evidence in this case, including review and analysis of any information that may be provided to me subsequent to the filing of this Declaration. I am prepared to testify about these opinions.

XI. CERTIFICATION

[267] I hereby certify and declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful and false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Dated: September 12th,2025 Respectfully Submitted,



Edwin A. Hernandez. Ph.D.