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 IPR2025-00351

U.S. Patent No. 11,140,441

DECLARATION OF DR. EDWIN A. HERNANDEZ-MONDRAGON

TABLE OF CONTENTS

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. 11,140,441 ("the '441 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.

[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. 11,140,441 (Ex- 1001)
- Claim Construction Briefs filed as part of S.D. of Florida litigation (EX2052)
- Deposition of Mr. Lipoff (EX2053, EX2054, EX2055)
- Ex-1002 and Petition for the '441 patent
- Exhibits referenced by Lipoff:

Table of Exhibits

Ex. No.	Description
Ex-1001	The '441 Patent.
Ex-1002	This Declaration.
Ex-1003	U.S. Patent No. 7,940,303 to Farber, et al. ("Farber"), which was filed on November 12, 2009.
Ex-1004	U.S. Patent No. 8,954,600 to Avellan, et al. ("Avellan"), which was filed on March 2, 2012.
Ex-1005	U.S. Patent No. 9,009,111 to Vermeulen, et al. ("Vermeulen"), which was filed on May 14, 2012.
Ex-1006	U.S. Patent No. 8,442,110 to Pavlovskaia, et al. ("Pavlovskaia"), which was filed on October 9, 2008.
Ex-1007	WIPO International Publication Number WO 2010/044926 to Pavlovskaia, et al. ("Pavlovskaia-PCT"), which was published on April 22, 2010.
Ex-1008	U.S. Patent No. 8,676,822 to Davis ("Davis"), which was filed on February 6, 2009.
Ex-1009	U.S. Patent Application Publication No. 2012/0117569 to Fogel, et al. ("Fogel"), which was published on May 10, 2012.
Ex-1010	U.S. Patent No. 8,819,043 to Durante, et al. ("Durante"), which was filed on November 9, 2010.
Ex-1011	U.S. Patent No. 9,635,075 to Ma, et al. ("Ma"), which was filed on March 8, 2010.
Ex-1012	U.S. Patent No. 6,463,445 to Suzuki, et al. ("Suzuki"), which was filed on September 30, 1999.
Ex-1013	U.S. Patent Application Publication No. 2004/0031052 to Wannamaker, et al. ("Wannamaker"), which was published on February 12, 2004.
Ex-1014	U.S. Patent No. 9,331,967 to Gangadharan, et al. ("Gangadharan"), which was filed on October 31, 2013.
Ex-1015	Prosecution History of Application No. 15/538,911, which is now U.S. Patent No. 10,123,074.
Ex-1016	Prosecution History of the '441 Patent.

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 '441 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
- 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 have licensed other portfolios of my patents to Verizon, Cisco, Dish Wireless and other major technology players 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 '441 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 '441 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.

Level of Ordinary Skill in the Art

[18] 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

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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

[25] 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

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

IV. SUMMA∂RY OF OPINION

[34] The Petition sets forth 13 grounds:

Table 1. Grounds of Invalidity

Ground 1: Claims 1 and 7 are anticipated by Farber
Ground 2: Claims 1 and 2 are obvious in view of Farber and Vermeulen
Ground 3: Claim 3 is obvious in view of Farber and Pavlovskaia
Ground 4: Claims 4, 5 and 8 are obvious in view of Farber and Davis
Ground 5: Claims 6 and 9 are obvious in view of Farber and Fogel
Ground 6: Claims 10-16 and 18-23 are obvious in view of Avellan and Pavlovskaia-PCT
Ground 7: Claim 15 is obvious in view of Avellan, Pavlovskaia and Durante
Ground 8: Claim 16 is obvious in view of Avellan, Pavlovskaia-PCT and Ma
Ground 9: Claim 17 is obvious in view of Avellan, Pavlovskaia-PCT and Suzuki
Ground 10: Claim 24 is obvious in view of Avellan, Pavlovskaia-PCT and Gangadharan
Ground 11: Claim 25 is obvious in view of Avellan and Wannamaker
Ground 12: Claim 26 is anticipated by Avellan
Ground 13: Claim 26 is obvious in view of Avellan and Wannamaker

[35] Claims 1-26 are the challenged claims. Claims 1, 10, 25, and 26 are independent claims the other claims are dependent.

[36] Mr. Lipoff relied on the petition in his analysis and did not conduct an independent analysis (Lipoff Deposition EX2053 at Pg. 144, lines 2-6, Pg. 116, lines 15-20, 23-25, Pg. 117, 1-8 and 19-23). For example:

[37] Therefore, Lipoff *only* provided analysis in his declaration for some not all claim elements for instance for Claim 1 only addressed 1b, 1d, and 1h

[38] However, I have reviewed the Petition and Declaration to form my opinions.

[39] Ground 1 relies on Faber for Claim 1 and 7, this fails as Farber is a machine that uses "triggers" to transition on state to the next, all the way to multiplexing using a "MUX". At the MUX, then multimedia generated as an MPEG Transport Stream is stored. Hence, as one process starts other finishes and, is very time driven. Farber cannot anticipate the '441 Patent as, the entire patent is based on web-services and stateless, not stateful. Hence, if a request is received in any of the states of Farber it will disrupt the process of generating MPEG TS. Besides that, if the Multiplexer is also used to transmit the content to the Cable Operator, Elements 1[g] and 1[h] are not needed and in fact will cause Farber to start generating more content, that is already broadcasting. Farber's FIG 6 cited by Petition, shows a template that drives the system, however such XML does not show any "multimedia file" reference, instead points to an XML tag "CD Value" indicating that, Farber does not operate with "music" files at all.

[40] Additionally, Farber teaches that the "Image Encoder" and "MPEG Encoder" generate "MPEG Transport Streams" which are processes used for "Encapsulation" not "encoding," Lipoff in his declaration altered the way he refers to "**MPEG Encoder"** and switched it to "**MPEG Video Encoder"** Ex1002¶146 and ¶150. This change then to suggest the reader the existence of an "*video encoding*" module and that Farber also "*encodes*" video, when is disclosed that only works with "***MPEG***

Transport Streams". Besides that, Ex1002¶FIG. 6 shows that in the XML for a template, instead of showing a file for the multimedia (e.g. ***music.mp3***), Farber uses **"CD Value"** indicating a far different system than the Petition describes. Petition, 29 cites the template and fails to show any cloud or web resource, or how that could be added to Farber.

[41] Ground 2 relies on Farber and Vermeulen for Claim 1 and 2 and also fails for similar reasons as Ground 1. However, in determining space in a multimedia system needs to be done based on a Threshold, as modern File Systems, given i-Node limitations will return space available, when there will not iNodes available for multimedia causing corruption and crash of a system

[42] , Ground 3 relies on Farber and Pavlovskaia for Claim 3, as such as Claim 1 is not met, then this is not met, as Farber does not disclose how the "Image Encoder" converts HTML to MPEG Transport Streams. Hence, the HTML changes illustrated by Pavlovskaia. However, a RESTful system from Pavlovskaia that is stateless is not combinable with a stateful system as in Farber.

[43] Ground 4 relies on Farber and Davids for Claim 4, 5, and 8. As Davids introduces new codecs, Davis will require modification of the "Image Encoder" and "MPEG Encoder" and possibly the "Audio Encoder" that not known. Therefore, this combination will require undue experimentation.

[44] Ground 5 relies on Farber and Fogel for Claims 6 and 9, as “Image Encoder” is unknown, then not only the “Image Encoder” will require modifications to receive an HTML page and cover it to an MPEG Transport Stream of the web page.

[45] Ground 6 relies on Avellan and Pavlovskaia-PCT for Claim 10-16 and 18-23, Pavlovskaia-PCT offers a single element that reads “Virtual Machine” as s component of a system that uses it. As claims related to virtualization are supported by the specification as in FIG 10, 11 and others. Petition does not show how the teachings of Pavlovskaia could lead any virtualization in Avellan. Besides, that Avellan suggest the use of Java as part of a Java Applet that is remote on the web page, and does not suggest that Avellan can be virtualized, having a “satellite interface” that will have be virtualized as well.

[46] Ground 7 relies on Avellan, Pavlovskaia, and Durante for Claim 15, for the same reasons as Claim 14, Claim 11, and Claim 10. Besides that, Avellan relies on a “images” captured from “a web page,” The Petition claims that an “image captured” that Avellan turns into compressed files with the “Web to HD Video Protocol” and just because the captured web site is a “media web page”, such “CNN, a Live Stream, Youtube, etc”

[47] For the grounds relying on Avellan. 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"

[48] 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.

[49] 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.

[50] 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)

[51] 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.

[52] 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.

[53] 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

[54] 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.

[55] .

[56] Ground 8 relies on Avellan, Pavlovskaia-PCT and Ma for Claim 16 ,hence the same reasons for Claim 10, Besides all the issues in Ground 7. The disclosure in

Pavlovskais-PCT is insufficient so suggest that Avellan can be properly virtualized having a “satellite link” being part of Avellan’s system, as well as modification to the “remote users”, contrary to the ‘441 Patent FIG 10 and FIG 11.

[57] Ground 9 relies on Avellan, Pavlovskaia-PCT and Suzuki for Claim 17, However a combination with Suzuki suggest that the “Content Providers” are the users not the Gateay device, plus Pavlovskais-PCT is insufficient so suggest that Avellan can be properly virtualized having a “satellite link” being part of Avellan’s system, as well as modification to the “remote users”, contrary to the ‘441 Patent FIG 10 and FIG 11.

[58] Ground 10 relies on Avellan, Pavlovskaia-PCT and Gangadharan, plus Pavlovskais-PCT is insufficient so suggest that Avellan can be properly virtualized having a “satellite link” being part of Avellan’s system, as well as modification to the “remote users”, contrary to the ‘441 Patent

[59] Ground 11 fails per Avellan and Wannamaker for Claim 25, Avellan and Wannamker are competing solutions a) Avellan uses image captures from web pages, and b) Wannamaker defines Rendering as “Server Transcoding” and converting HTML pages into FBML/MML Pages with FBM images for a microbrowser. Hence, there will be the browsers in Avellan competing with the microbrowser at Wannamaker.

[60] Ground 12 fails per Avellan for Claim 26, as Petitioner points to Avellan to have a "multicast unit" as cites the use of "multicast groups" however Avellan,

[61] Ground 13 fails per Avellan and Wannamaker for Claim 26, as Avellan and Wannamaker cannot be combined due to competing technologies, the arguments made on the redundancy attributed to be fault tolerance and multiple servers, are not faults related to "providing at least a portion of the recorded video," as this content will be provided as faults such as "absence of the multicast address," "ping response timeout," and "streamer" process execution tracking. Besides that Wannamaker's OC1000 Motorola device that works with the MMS platform cannot generate a multicast only receive it, for later sending to the QAM Modulator.

V. PERSON OF ORDINARY SKILL IN THE ART

[62] A POSITA with respect to the '441 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.

VI. OVERVIEW OF THE '441 PATENT

[63] The '441 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.

[64] The '441 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.

[65] 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.

[66] One novel aspect of the '441, 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

‘441 Patent at 5:54-63

[67] In the ‘441 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.

[68] Consequently, the ‘441 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.

[69] The ‘441 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.

[70] Additionally, the ‘441 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 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

[71] 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 '441 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

[72] 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 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 webpage contains a news or stock ticker:

The result of using the '441 is a broadcast media stream with the "rendered web page" as shown here:

B. The MediaPlug

[73] As explained before, the methods and systems disclosed by the '441 patent are stored in the "MediaPlug" device (See *Id.* at FIG.1 Element 110 and FIG. 2).

[74] The '441 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. The MediaPlugs teach the use of a "caching unit" and a "multicasting unit" that generate the media streams and handle multicasting streams or HTTP Live Streaming (EX2056)

C. Playlists and MIME Types

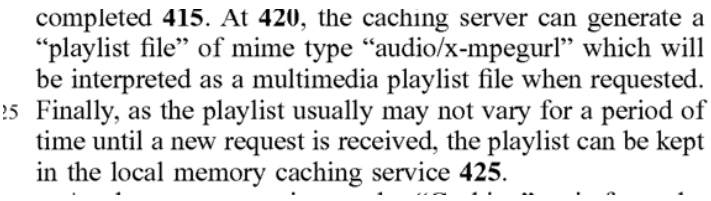
[75] Further, the '441 patent teaches that such playlist is stored as part of an internal web server are processed in response to an HTTP Request where the web server changes the HTTP "header" information of the response to be announced to a requestor, e.g. a Web client that, a playlist is in use. As specified in the RFC2112[1],

[1] https://datatracker.ietf.org/doc/html/rfc2112 See EX2063

the protocol provides a "MIME-like message" containing information on the request made.

[76] 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)

[77] 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

[78] 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)

[79] Multimedia container formats MUST not be confused with a video encoding formats, as some containers can use certain encoders, as shown in the '441 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. (EX2042, 27-83)

[80] Therefore the '441 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

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

[82] 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 (See EX2042, 92-97)

[83] Hence, the ‘441 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)

[84] In summary, the ‘441 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 ‘441 Patent.

VII. CLAIM CONSTRUCTION

[85] 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

[86] 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"

[87] 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"

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

H. Web Page

[89] 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

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

[91] "Render a Web Page" is described by the '441 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.

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

VIII. PATENTABILITY ANALYSIS FOR CHALLENGED CLAIMS

[93] 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 '441 Patent.

[94] The "Anticipation" arguments disclosed by Petitioner fall into these categories:

a. Anywhere on Farber elements can receive a request

b. Coding by hand an HTML interface satisfy as a method claim

c. A stateless protocol such as HTTP is combinable with a "state machine" system,

d. Any of Faber Elements has a storage and each claim element, "caching unit" can be moved around,

e. 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."

f. For the "Farber" reference, the "caching unit" is Element 140 or "System 140" (See. EX2055 , at 47:24-25, 48:1-7) at Ex1003 at FIG 5 is the "caching unit" and for some other claims the "storage device" is either Elements 42, 44 and/or Element 150

[95] The first set of grounds are based on "Farber." Farber Ex1003 teaches two embodiments one for broadcasting and one for storing the content. Farber discusses how each element in *Id.* FIG 4 is used for the "broadcast" at text below MUX 48 (Multiplexer), in the second embodiment, Farber stores from the MUX 148 into a Hard drive MUX 150. Farber teaches that it can "It should also be understood that the system 140 is capable of simultaneously transmitting, playing out, or storing such content " *Id.* 8:23-25.

[96] The analysis of each ground is presented herein:

A. Ground 1 – Farber fails for Claims 1 and 7

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

[98] A computer-implemented method comprising:

a. creating a plurality of multimedia assets at a caching unit, for each one of the plurality of multimedia assets:

b. determining that there is sufficient storage space for the multimedia asset at the caching unit;

c. generating a channel identifier for the multimedia asset;

d. retrieving from a cloud service a plurality of media files to be included in the multimedia asset including a media identifier for each one of the media files;

e. for each one of the media files, creating a custom hypertext transfer markup language (HTML) user interface that includes video;

f. encoding each of the media files and encapsulating them together using an MPEG transport stream format;

g. storing each one of the multimedia assets at the caching unit;

h. receiving a request at the caching unit for one of the multimedia assets from a broadcasting unit; and

i. responsive to receiving the request, the caching unit providing the multimedia asset to the broadcasting unit in the MPEG transport stream format...

[99] Claim 1 recites Element 1[a] "creating a plurality of multimedia assets at a caching unit, for each one of the pluralities of multimedia assets", and, Element 1[e] "for each one of the media files, creating a custom hypertext transfer markup language (HTML) user interface that includes video;"

[100] Petition, 12 indicates fails to show that such "for each of the media files, creating a custom hypertext transfer markup language (HTML) user interface that includes video " and point to Ex1003 FIG 1 as evidence.

[101] Petition, 13 says that "May be coded using HTML" at Petition, 13. *First*, This statement indicates that a manual process is used "may be coded," *Second*, "fore reach of the media files…" indicates an automated process that is part of "creating a plurality of multimedia assets" hence needs software.

[102] The '441 Patent discloses the use of an HTTP Server, PHP-based, Python, or NodeJS or another HTML User Interface generators, as follows:

a. The '441 teaches that "HTTP Server may run PHP or Python" at Ex-1001,9:45. This server "Creates the User Interfaces"

b. On Ex-1001 at *Id.* 6:65-67 and 7:6-8 in reference to "can add a "Custom UI" or customizable User Interface …. or a **web service**"

c. A web service is executable request/response mechanism based on the REST protocol that retrieves HTML that is executed by the web service at the HTTP Server.

d. This web service provides the template, via an HTTP GET request.

[103] Therefore, Farber only discloses the use of "Templates" that could be simple HTML formatted or XML formatted files, not "User Interfaces that includes Video."

[104] For a "User Interface that includes videos" implies the use of an HTTP Server or a RESTFul system to disclose Element 1[e], and Data Carousel does not disclose that.

[105] Moreover, given Farber's "Triggering" system, it will ***not*** suggest the use of web-services such as HTTP and any RESTUL APIs that are stateless. HTTP-based systems are stateless and present challenges of their own, but "triggered" systems are stateful.

[106] As shown the "Data Carousel" cannot perform "for each one of the media files, creating a custom hypertext transfer markup language (HTML) user interface

that includes video;" because simply it cannot serve it via an HTTP Request. Data Carousel it needs a "trigger" to be processed "for each one of the media streams," from the "Audio Encoder 49,"

[107] Since a POSITA will now that State machines, by definition, rely on maintaining internal state across transitions, where each input triggers a state change based on the current state. In contrast, HTTP-based web services (especially RESTful ones) are designed to be stateless: each request must contain all necessary information to be processed independently, without the server retaining memory of prior interactions.

[108] As a consequence, the "Data Carousel" first receives a "Trigger" from the "Audio Encoder 49 and, at a pre-determined time, sends a trigger to the data carousel subsystem 46 to initiate an associated data feed"

The broadcast playout system **50** is the mechanism by which fundamental programming content is played from recorded media. It is often based on commercially available broadcast automation hardware and software. The broadcast playout system **50** sends audio content (e.g., an audio work), such as a queued song, to the audio encoder **49** and, at a pre-determined time, sends a trigger to the data carousel subsystem **46** to initiate an associated data feed. The data

Ex1003 5:60-67

[109] Clearly, the data carousel cannot use HTTP, if it did the "trigger" from the Audio Encoder 49 would receive as a response as part of the HTTP Request, but instead is not expecting a response, but another trigger from the previous state.

[110] In other words, Farber's "Data Carousel" or the "associated data feed," will be returned as part of the "trigger" and the entire system would have to be changed. Instead such data feed generates another trigger to the MPEG encoder 47 (See *Id.* at 6:1-3).

[111] Hence, when the Petition, 14 says that these "User Interfaces" that are part of the "multimedia assets" are referenced by an "identifier," for example "Soft Rock" as in Ex1003, Fig. 6, such identifier is present but for Element 1[d] such "Identifier" is used for retrieval for Element 1[d] or "Retrieving from the Cloud service provider….including a media identifier." Petition points to FIG 4 Data Sources 44, 46 to be the "Cloud service" at Petition, 21. Element 1[d]. Therefore, Data Sources 44 to 46 cannot be the "Caching unit," as usually a cloud-service provider will not be suitable to "create a multimedia asset that is being encoded." Although, this could be true, Petition 21-22 has not shown that the media identifier is used for retrieval. This is an important feature, as playlists change and music royalties are tied to those playbacks, keeping track of those identifiers is fundamental. Simply, Petitioner shows as evidence, where the XML includes "CD Value" and "UniqID," given that Farber priority date is March 7, 2003. This is right before the Apple launched

iTunes in 2003 that began with the use of digital formats, such as MP3s. Hence, it is likely that Farber and the data sources are CDs not digital files, as shown there is no media file. For example for a cloud-based service: http://s3.amazon.com/..../song.mp3 or a local drive: "c:\music\file.mp3" or "/music/file.mp3," in Farber below it could be **https://s3.amazon.com/../3310722.mp3.** Petition 21, points to ".jpg" artwork, but that is a local file, and a ".jpg" image is not media file. (*Id.* FIG 6)

```
<Grobj
xPos="214" yPos="92" xSz="214" ySz="251" wid="214" hgt="251"
pos="1" >
 <SongTitle value="Are You Ready For Love (2003)"/>
 <Artist value="Elton John"/>
 <RecordLabel value="ULTRA"/>
 <CD value="Elton John&amp;apos;s Greatest Hits 1970-2002"/>
 <UniqID value="3310722"/>
</Grobj>
<Grobj src="mcBT_BuyPrevious_1.jpg" desc="Buy Previous Button"
xPos="209" yPos="314" xSz="129" ySz="37" wid="129" hgt="37"
```

[112] Therefore, by no means Claims 1[a], 1[b], 1[d], and 1[e] are disclosed by Farber.

[113] In fact, for Element 1[a], Farber teaches away from "creating a plurality of multimedia assets at a caching unit, for each one of the pluralities of multimedia assets," as the data sources are used as references, and each state passes "items" with each "trigger"

[114] Faber further discloses, at *Id.* FIG 4 that the entire FIG 4 and FIG 5 is a "triggered" based machine that requires "real-time" coordination to match, each template with the songs in the Audio Encoder 49 from *Id.* 5:65 that are multiplexed with MPEG Encoder 47 into (*Id.* 6:18-21) upon a trigger from the Data Carousel *Id*. 6:1-3. However, at Id. 2:64-67. Farber explains that its entire system is "based upon a trigger"

"screen", which is subsequently output to a display. Based
65 upon a trigger, the assets are gathered again and reassembled based on a second template to form a second video frame or "screen" wherein the assets are in positions on the screen,

Ex1003 at 2:64-67

[115] Hence, Farber cannot disclose "determining space" for the "creation of multimedia assets," as such task is unnecessary, especially when CDROMs are in use. In fact, Element 1[b] will require special experimentation as a) Each state will have to determine its own space, such that the "trigger" mechanism works successfully, and b) What is total, when many "general-purpose" computers are in use as in Farber.

[116] Therefore, Element 1[b] "determining that there is sufficient storage space for the multimedia asset at the caching unit," at best would be "determining space per state per trigger" or after each trigger. Additionally, as *Id.* 2:64 says, "Based upon a trigger, ***the assets are gathered again and reassembled*** …." indicates that there is no need for "Creating multimedia assets" with Farber.

[117] Hence, Farber requires each step to happen as specified after each "trigger" and in the order shown in FIG 4 and FIG 5. The Multiplexor or MUX results that are stored at the "storage/playout" Element 150 of FIG 5 occurs after the "audio encoder" triggers and/or the "MPEG Encoder" is ready to deliver the content to the multiplexer. At that moment, the "MPEG Transport Stream" is ready and starts being "Broadcasted" or "provided" to the Cable TV headend immediately after being multiplexed (Petition, 25 and Ex1003 at 6:21-25). Hence, the MPEG transport stream have been delivered to the "Broadcasting Unit." This behavior teaches away in Farber, for the use of Elements 1[h] and Element 1[i]. Since, the Multiplexer has already delivered "music and graphics" to the Cable TV system, there is no need for Element 1[h] "receiving a request at the caching unit for one of the multimedia assets from a broadcasting unit" and Element 1[i] "providing the multimedia assets to be provided to the broadcasting unit" as the "Broadcasting unit" .

[118] Farber requires triggers because each device in *Id.* at FIG 4 and FIG 5 is a "hardware-based system" or an arrangement of "server devices" that will require "undue experimentation" by a POSITA to handle coordination for "receiving a request" or Elements 1[h] and 1[i]

[119] Hence, Farber does not disclose a "one unitary system," or a "Caching unit" as Lipoff pointed in "Lipoff Deposition Day 2" EX2055, 49:8-24 when he established that Element 140 was the "Caching unit"

[120] Clearly, a "caching unit" is a "unit" for Element 1[h] to happen "receiving a request at the caching unit." Even if Element 140 (or System 140) is the caching unit, Farber will have to show some coordination within "an asynchronous request calls and coordination with triggers" in Farber's system, as the Petition, 12 recites.

> Thus, Farber's disclosure of a "music channel" that includes encoded audio data and corresponding MPEG video frames for the queued song/channel discloses a "multimedia asset." As detailed below, each music channel is associated with a "channel identifier" and includes a "plurality of media files"—such as songs, album artwork and video assets—that are encoded in an "MPEG transport stream" and cached for subsequent playback. This process "may be executed for a ***plurality of channels*** simultaneously." Ex-1003 at 6:9-10. Accordingly, Farber discloses "creating a plurality of multimedia assets at a caching unit," and "for each one of the plurality of multimedia assets" Farber further discloses the elements below.

Petition, 12.

[121] If Farber is generating, and recording, Farber cannot receive a request, as the state machine and all "Triggers" will fail.

[122] However, Petitioner does not point to the "Caching unit" with any precision, whether is the "Data Carousel" Element 146 at FIG 5 or "Broadcast Playout System" Element 50 at FIG 4, or the MUX's Storage Device 150 at FIG 5 "caching unit." Despite, Lipoff statements and Petition,

[123] Again, a trigger-based state machine, cannot receive a "Request from a caching unit" as it will disturb and will break the flow of the triggering mechanics.

[124] Cleary, Stateless and Statefull systems are incompatible.

[125] Besides that, Farber system is composed by many components that have storge and that are "implemented" as a "Personal Computer or a "General-Purpose Computer", having associated storage capabilities" each of those:

a. Data Carousel, Element 146 (See *Id.* 5:40)

b. MPEG Encoder, Element 47 (See *Id.* 6:14)

c. Image Encoder, Element 147 (See *Id*. 7:40)

d. Audio Encoder, Element 149 (See *Id.* 7:50)

[126] Therefore, all elements in Farber have "Storage" devices. Besides, Element 150 at FIG 5 (See Petition, 16) that is a storage device, and follows at Pg. 17 with:

> POSITAs would recognize that the successful playout of the requested media stream is a "determin[ation] that there is sufficient storage space" for the multimedia asset at the caching unit. Ex-1002, ¶126. In other words, by retrieving "at a later time" the stored data, there must have been sufficient storage space for that data. Ex-1002, ¶126.

[127] Given the number of storage devices, is then such amount to a "a specific set of disks" that are only used for storage or just to Element 150? if Farber is a stateful system, each state will have to know the "space available"

[128] Therefore, Lipoff testimony contradicts the Petition, 17 that the "caching unit" is Element 140 versus the Petition Element 150 (in red) at Petition, 16 from Ex1003 FIG 5.

[129] Additionally, Element 1[b] in the "Determination of Available Space" is a crucial step any "Multimedia System" especially those that generate multimedia assets.

[130] As all software is stored in the same server and storage file system, a POSITA will now that certain amount of storage is used by a server's File System for:

a. Operating System Files (e.g. Linux) and Swap
b. Software baseline, e.g. FFMPEG, VLC, Python, or PHP
c. The music files database from a source,
d. Log files and,
e. The space to be used for the generated multimedia assets to be created.

[131] In some systems, there is no difference between retrieving available space before or after a process, in multimedia streaming this is not the case, especially as the one in the '441 Patent:

a. *First,* when the media encoder, e.g. FFMPEG starts encoding any video into a certain bitrate, if the process is not completed, the file may present multimedia related errors such as "MOOV atom not found" error[2]

b. *Second,* in many cases encoding of a file may take place real-time, hence if a file length is X minutes, the time to encode is close to that time,

c. Third, lack of space will turn the file unusable and will corrupt an entire database of media files

[132] Perhaps, when hardware encoders are used and dedicated hardware handles multimedia processing, failure to have space can be handled at the "Digital Signal Processor" otherwise, lack of space will generate corrupted outputs and difficult errors to resolve.

[133] In a stateful and triggered-based system, storage is not an issue, as there are many storage units as in Faber. Besides that, as one process completes after the other

[2] https://stackoverflow.com/questions/55896329/how-to-fix-moov-atom-not-found-error-in-ffmpeg

start, deleting old resources is not an issue, as data is passed to the next process that is triggered..

[134] In contrast, the main issue is in Linux filesystems with "Multimedia files" as in the '002 Patent is that a large file, specially occupying 100MB or several Gigabytes per multimedia asset requires some attention, as, *i*-Node capacity for large files needs to be considered. A filesystem device may run out of *i*-Nodes and when this happens, even though there may be "free space available," and error will occur to save a file (EX2049, 3).

> It is possible for a device to run out of inodes. When this happens, new files cannot be created on the device, even though there may be free space available. This is most common for use cases like mail servers which contain many small files.
>
> Filesystems (such as JFS or XFS) escape this limitation with extents and/or dynamic inode allocation, which can 'grow' the filesystem and/or increase the number of inodes.

[135] Hence, determining a safe "threshold" to keep as reserved space is fundamental in multimedia systems.

[136] In general, Operating systems run tasks in parallel that will need sufficient space left to handle error logs, resources, thousands of small album art files, and other "File System" elements. A page fault or a ENOSPC[3] will occur if suffieciet amount of space is not left in the filesystem, such error could be fatal and potential system halt. If this occurs, in a remote location, sometimes is hard to recover a harddrive with i-Node errors or filesystem corruption. Hence a POSITA with software engineering experience will understand that as Lipoff describes and Petitioner indicates, is not a good software practice.

> A virtual memory scheme splits the memory used by each program into small, fixed-size units called *pages*. Correspondingly, RAM is divided into a series of *page frames* of the same size. At any one time, only some of the pages of a program need to be resident in physical memory page frames; these pages form the so-called *resident set*. Copies of the unused pages of a program are maintained in the *swap area*—a reserved area of disk space used to supplement the computer's RAM—and loaded into physical memory only as required. When a process references a page that is not currently resident in physical memory, a *page fault* occurs, at which point the kernel suspends execution of the process while the page is loaded from disk into memory.
>
> On x86-32, pages are 4096 bytes in size. Some other Linux implementations

[137] In fact, Ex1002 ¶126 states that "storing and retrieving the multiplexed transport stream for subsequent playback" inherently teaches Element 1b, I disagree as the way presented in the '441 i

[3] **ENOSPC** No space left on device (POSIX.1-2001).

[138] As presented earlier since the "Image encoder 47 receives output from the data carousel subsystem 146 to create a video frame" *Id*. 5:37-42, such data is no longer needed in the "Image Encoder 47" module, and determining space available doesn't make sense, as that "data" can be just deleted and only stored in "Storage / Playout 150" at FIG 5.

[139] Hence, Farber and Element 1[b] is not present as Farber's configuration is a series of servers.

[140] Petition, 22 1[e] again **"In response to the trigger,** the encoder 47 pulls template information… from data carousel." Ex1003 6:2-7

carousel subsystem **46** generates a trigger to an encoder **47**, which may be an MPEG encoder **47**. In response to the trigger, the encoder **47** pulls template information, images, and text from the data carousel subsystem **46** and creates, based on the template information, a screen having embedded assets such as those shown in screens **1**, **2** or **3** which are associated with the currently queued song. The screen may be a video frame, such as an MPEG video frame.

[141] For Element 1[f], Farber requires to show that "encoding each of the media files and encapsulating them together using an MPEG transport stream format;"

a. First, "***Encoding*** each of the media files, and"

b. Second, "and ***encapsulating*** them together using an MPEG transport stream format"

[142] In other words, each of the media files is encoded and encapsulated. At best Farber only discloses encapsulation at the MUX 150. Encoding means using a video and audio encoder, and encapsulation means for example, using an MPEG Transport Stream.

[143] Farber discloses "MPEG Transport Streams" only:

a. First, Petition, 25 cites Ex1003 at 6:15-18, "Multiplexer 48 receives the MPEG Transport Stream from the MPEG Encoder 47" what this means is that the MPEG encoder returns a "Transport Stream" only,

b. Second, Petitioner combines embodiment FIG 4 with FIG 5 and sys that "As the "Audio Feed 149 supplies audio with is associated with the video frame output of the video encoder at 147" Ex1003 at 7:43-45.

[144] Regardless, the Multiplexer 48 always returns an "MPEG Transport Stream," which means is that "Audio Encoder" also returns "MPEG Transport Stream," as the "MPEG encoder" also returns "MPEG Transport Stream."

[145] MPEG Transport Streams are "Encapsulation" Formats not encoding formats.

[146] This is further confirmed by Farber at *Id.* 6:9-25

It should be understood that this process may be executed for a plurality of channels simultaneously. The MPEG encoder **47** then creates an MPEG transport stream for all channels with the embedded MPEG video frames. The MPEG encoder **47** may be implemented utilizing a commercially available encoder or a general-purpose computer.

Based upon a trigger from the broadcast playout system **50**, a multiplexer **48** receives the MPEG transport stream from the MPEG encoder **47** and simultaneously receives an encoded audio feed from an audio encoder **49**. The multiplexer **48** serves to combine the audio feed from the audio encoder **49** and the associated encoded MPEG transport stream from the MPEG encoder **47**. The multiplexer **48** then feeds the multiplexed signal out for broadcast in the form of a video transport stream to a distribution system, such as the head end of service provider (e.g., a cable television network or a satellite network provider as is well known in the art).

Ex1004 at 6:9-25

[147] Additionally, Farber discloses the "audio encoder" returns an "MPEG Transport Stream" that is Multiplexed with the video from the "MPEG Transport Stream".

[148] Given that only "encapsulation" is disclosed by Farber and no encoding. As shown, only "MPEG Transport Streams" references are made, and as Element 1[f] language reads ***"encoding*** each of the media files and ***encapsulating*** them together using an MPEG transport stream format " (emphasis added)

[149] Lipoff changes the "text" of the "MPEG Encoder" in FIG 4 to read "MPEG Video Encoder" at Ex1001 ¶146, to pretend that "Encoding and Encapsulation" are taking place, and imply that Farber "Encodes and encapsulates," when Farber ***only*** operates in "MPEG Transport Streams"

146. POSITAs would have had a reasonable expectation of success in combining Farber with Davis's disclosure of a specified bit rate because Farber already contemplates use of an MPEG video encoder and an audio encoder and assembling media into streams for delivery and the specified rates of Davis could readily have been used to encode the audio and video data of Farber. Encoding data

Ex1001,¶146

[150] This is clearly a misleading statement, despite Farber describing that the "MPEG Encoder" work ***only*** on "MPEG Transport Streams" and does not encode anything.

[151] Lipoff repeats his statement at ¶150 to combine it with David's.

150. POSITAs would have had a reasonable expectation of success in combining Farber with Davis's disclosure of H.264 because Farber already contemplates use of an "MPEG" video encoder and H.264 is an MPEG video encoding standard—i.e., MPEG-4 Part 10. Ex-1003 at 6:2-7.

[152] And, then cites, Ex1003 at 6:2-7 as a reference, that by no means makes any reference to any "MPEG Video Encoder" language.

carousel subsystem **46** generates a trigger to an encoder **47**, which may be an MPEG encoder **47**. In response to the trigger, the encoder **47** pulls template information, images, and text from the data carousel subsystem **46** and creates, based on the template information, a screen having embedded assets such as those shown in screens **1**, **2** or **3** which are associated with the currently queued song. The screen may be a video frame, such as an MPEG video frame.

Ex1003 6:2-7

[153] Notwithstanding Petitioner efforts to recalibrate and reframe what the "MPEG Encoder" and rename it to "MPEG Video Encoder."

[154] Switching a name implies that Petition and Lipoff are flawed in their analysis of Farber and their mix and match media encoding with media encapsulation.

[155] . Another flaw but on Farber's disclosure is that it is unable to teach how the "HTML" is converted to an "MPEG Transport stream" using the "Image Encoder" from FIG 4 or FIG 5 into an "MPEG Transport Stream,"

[156] Farber limits itself to say that "Image encoder 47 receives output from the data carousel subsystem 146 to create a video frame" *Id*. 5:37-42, without disclosing how this is done. In contrast to what the '441 patent description describes using PhantomJS and FFMPEG.

The image encoder **147** receives output from the data carousel subsystem **146** to create a video frame. The image encoder **147** may be implemented within a personal computer or general purpose computer or may alternatively be implemented as a separate piece of encoding equipment which is commercially available for generating encoded video frames.

[157] Additionally, Farber limits itself to say at *Id.*6:9-14, that the "MPEG Encoder 47 maybe implemented utilizing a commercially available encoder or a general-purpose computer"

It should be understood that this process may be executed for a plurality of channels simultaneously. The MPEG encoder **47** then creates an MPEG transport stream for all channels with the embedded MPEG video frames. The MPEG encoder **47** may be implemented utilizing a commercially available encoder or a general-purpose computer.

Ex1003 6:9-14

[158] Clearly, an MPEG video frame should exist and be part of a "Program Identifier" as part of the "MPEG Transport Stream," but the MPEG Encoder 47, in Faber, only outputs "MPEG Transport Streams" and receives "MPEG Transport Streams" as inputs.

[159] In fact, Farber teaches that if there is no audio in a feed, the output of the Image Encoder (also an MPEG Transport Stream) can be delivered to the cable or satellite operator (See *Id.* at 7:57-61)

A multiplexer **148** serves to combine the audio feed/encoder output **149** with the image encoder output **147** to create a transport stream at its output. It should be understood that the audio encoder **149** and the image encoder **147** may be triggered or otherwise timed to send output to the multiplexer **148** simultaneously in order to match desired audio with desired image content. It should also be understood that the multiplexer **148** may be removed from the system **140** when the optional audio encoder **149** is not used. In this case, the image encoder **147** output could be fed directly into the storage/playout device **150** which will be described below.

Ex1003 at 7:51-61.

[160] Therefore, Element 1[g] is not shown by Farber and teaches away from storing each of the multimedia assets one by one, except for storing the MPEG Transport Stream at the Multiplexer output, as one file into *Id.* Element 150 at FIG 5.

[161] As Element 1[g] requires "storing each one of the multimedia assets at the caching unit;" what this meaning is that for the Multiplexer for being able to generate a "Headend" compatible feed, it must be a continuous feed. As shown before, Farber first multiplex, then stores in the "Storage/Playout" at element 150 (The hard drive),

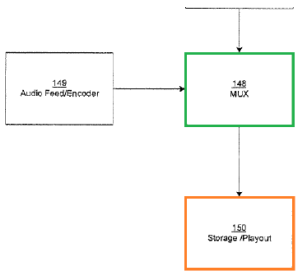


Fig. 5

Ex-1003 at Fig. 5. "[I]n applications where playout is desired at a later time, the storage/playout device 150 may be implemented utilizing a recorder for creating or writing to any suitable storage medium such as DVD, CD ROM, hard disk, or any other suitable read only or rewritable storage medium." Ex-1003 at 7:62-8:3.

Accordingly, Element 1g is disclosed by Farber.

See Petition, 26

[162] I disagree with Lipoff's opinion that the Multiplexer stores "one file at the time," (EX2055, 60:1-12),.

[163] Simply, the Faber discloses that the "The storage/ playout device 150 could also be a **direct playout device** such as a **display.** " *Id.* at 8:4-6 (emphasis added) If multiple files are stored in the same "Storage playout" Element 150 such output could not be a "direct playout" such as display, as it will require a way to select or switch what "channel" out of multiple "channels," and as shown Farber's motivations is to "solve the CRT / Screen burn-in" problem. Hence, a POSITA will know that Farber will be motivated to use a "Direct playout to a single CD/DVD/Hard disk" per Channel, rather than creating multiple channels in one storage unit, as in that arrangement a "direct playout device such as a display" would have been discarded.

[164] A POSITA will that for a display to convert the playout directly and in real-time, hence must be a single stream or a single file that is stored at device 150, not multiple files, otherwise a display will require the "playlist" to be known, and provided.

[165] Therefore, Farber does not disclose the use of a playlist as part of the "playout" device. Nor the Petition, 28-29 describes any "playlist"

[166] For example, a single file would be for example:

a. A single file would be: playout.ts,

b. A playlist will be a text file with the file names: playout-01.ts, playout-02.ts, playout-03.ts, …. Playout-*N*.ts, which will require a playlist of files to be loaded one by one.

[167] Now, element 1[h] language states that: "receiving a request at the caching unit for one of the multimedia, assets from a broadcasting unit; and"

[168] In general, this request at the caching unit is used to retrieve a "playlist" of media streams. I disagree that POSITA will know that "would recognize that such a request is received from the broadcasting unit. Ex-1002, ¶128. This mechanism is not disclosed by Petition nor by Farber.

[169] Therefore, as the Petition, 45, states the "caching unit" is Element 150,

> 1003 at 8:22-25. Thus, a POSITA would understand that the storage embodiment of Figure 5 would utilize the same methods as discussed with respect to the other figures of Farber, including Figure 4 and the accompanying discussion of communication with a broadcast distribution system.

[170] As clearly, the Petition, 27 and Ex-1002 ¶128 assumes that "Triggers" constitute requests of the "Caching Unit"

[171] As a consequence, Element 1[i] language requires that "responsive to receiving the request, the caching unit providing the multimedia asset to the broadcasting unit in the MPEG transport stream format," however this already took place for Element 1[g] as in FIG 5, the output of the mux or multiplexer is stored, and such output is also used to send to the cable operator or satellite operator. SeeEx-1003 6:21-25

[172] Regarding, Element 1[h] as the language reads that "receives a request and responsive to that request, the Caching unit provide the multimedia asset to the broadcasting unit."

[173] However, the evidence indicates that at *Id.* Element 150 at FIG 5 is labeled as "Storage/**Playout,**" and in FIG 4 the Element 50 is labeled "Broadcast and **Playout** System." This means that FIG 4 is the one that will playout the "recorded"

media, again these elements only receive triggers, and a trigger is not a request as defined by the "intrinsic" evidence of the '002 patent.

[174] Second, Farber teaches that *Id.* 5:60-62 "the **Broadcast System 50** is the mechanism by which fundamental programming content is played from the recorded media" (Emphass added)

[175] As such, if the "receiving a request at the caching unit", would be to the "Broadcast System 50" not the "Caching Unit," as specified by Petition, 7. Hence, as Farber teaches that the audio content such as queued song …. to the audio encoder,… as depicted at *Id.* 5:60-67.

[176] Clearly, although the audio makes it ultimately to the Multiplexer for Broadcast, by no means this is "responsive to the request, the caching unit provide the multimedia asset to the broadcasting unit"

[177] At best, the "recorded media" as specified by Farber at *Id.* 5:62, follows the state machine and all the triggers. If instead, the "storage / playout" at Element 150, is then viewable with a "that is capable of displaying or transmitting video images" *Id.* at 63-67

[178] Additionally, Farber also teaches that "The storage/ playout device 150 could also be a direct playout device such as a display" *Id.* 8:6-8

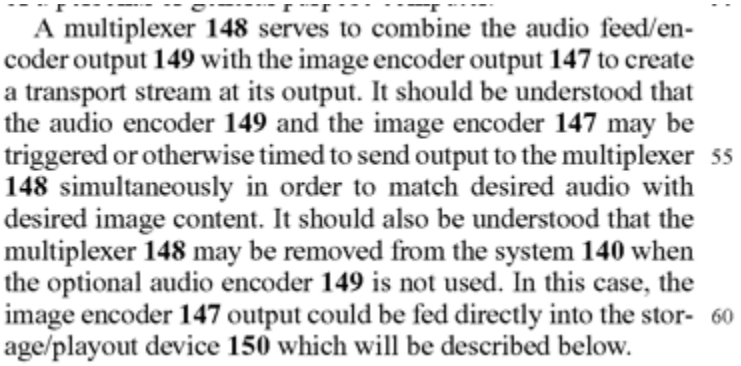
[179] Farber does not suggest by any means that there is a direct interface to the "storage/playout device 150" where a "broadcasting unit" can issue a request to receive the multimedia, and as I analyzed before, as a "request that is received by any element in Faber, will be unexpected, and the "triggering" mechanism will simply fail"

a. First, as the MUX 140 generates an MPEG Transport ready for distribution:

Fig. 5

b. Second, the Mux at 148 serves to "combine the audio feed/encoder output 149 with the image encoder output 147 to create a transport stream at its output" … additionally, "the audio encoder 149 and the image encoder 147 may be **triggered**

or otherwise timed to send output to the multiplexer" (**emphasis added**)



A multiplexer **148** serves to combine the audio feed/encoder output **149** with the image encoder output **147** to create a transport stream at its output. It should be understood that the audio encoder **149** and the image encoder **147** may be triggered or otherwise timed to send output to the multiplexer **148** simultaneously in order to match desired audio with desired image content. It should also be understood that the multiplexer **148** may be removed from the system **140** when the optional audio encoder **149** is not used. In this case, the image encoder **147** output could be fed directly into the storage/playout device **150** which will be described below.

c. This triggered and timed to send output to the multiplexer, as this is required to match the desired audio with the desired image content (from the Image Encoder) See *Id.* at 7:54-56

[180] Therefore, Faber teaches that such storage/playout device 150 is simply a "DVD Player" to which you can connect a "display," or a computer with a hard disk that can also have VGA or HDMI display connected to it.

[181] Therefore, Claim 1 is not anticipated by Farber.

1. Claim 7

[182] Regarding Claim 7 language is introduced herein "The method of claim 1, wherein the video for the custom HTML interfaces includes video taken from the media file."

[183] The Petition, 28 states that, Claim 7[a] that as Farber includes "motion picture video" at Ex1003 3:30-31. As explained before, Farber cites that the Data Carousel load "HTML" Templates that can include "Video" as part of the "Graphic object." However, Farber does not disclose how is this "Video" is loaded by the "Data Carousel,"

[184] As indicated in Claim 1 analysis, It is unclear, how this "Image Encoder" and "MPEG Encoder" work and how it can generate "MPEG Transport Streams" or is only capable of encapsulating content. How will it be able to generate from the "Template" the video output, as Farber limits the description at Id. 5: 37-42 and *Id.* at 9-14.

[185] Clearly, Faber simply states that "MPEG Encoder 47 may be implemented utilizing a commercially available encoder or a general-purpose computer" *Id.* at 9-14.

It should be understood that this process may be executed for a plurality of channels simultaneously. The MPEG encoder **47** then creates an MPEG transport stream for all channels with the embedded MPEG video frames. The MPEG encoder **47** may be implemented utilizing a commercially available encoder or a general-purpose computer.

[186] Regardless, of how this could have worked, the "Image Encoder 146 receive output from the data carousel subsystem 146" (*Id.* 5:37-38). Farber explains how ads and video is inserted into the stream, as follows at *Id.* 6:61-67

> In some embodiments, the template data described above may be generated by data carousel and provided to audio encoder 49 or MPEG encoder 47. Audio encoder 49 or MPEG encoder may place the template data within a "data pid" of the transport stream. This will allow a system at the head end of a service provider to retrieve the template data. Such a system

[187] In other words, Farber teaches that to submit ads or video, the "MPEG Endoder" sends as a part of the payload or "data PID" the template data. The Template data is then received by the headend and decoded. This is what Petitioner cites at Pg. 28.

> may use the information in the template data to effect local ad insertion. That is, the system may replace one asset included in the MPEG video frame created by and received from system **40** with another asset that is stored locally with respect to the system at the head end. A system for performing this local ad insertion function is described in U.S. Provisional Patent Application No. 60/623,246, filed on Nov. 1, 2004, the contents of which are incorporated herein by this reference.

Ex1003 at 7:1-8

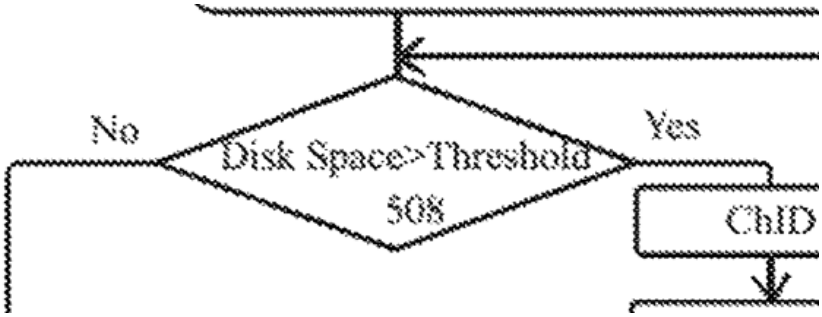
[188] Therefore, when petitioner cites "channel name 18, album art 20, title track label and artist information 22, [and] promotional/advertising panels 24, 28." Ex-1003 at 3:42-44. The template data is processed and videos are inserted in Farber as part of the "Multiplexing" step to the headend that will in turn insert the "Video advertisement:

[189] As a consequence, Farber does not disclose Claim 7.

B. Ground 2 - Farber and Vermeulen for Claim 1 and 2

[190] The Petition, 29 starts at Element 1[b] based on Vermeulen only and appears to rely on Farber for Clam 1[a] as analyzed in Ground 1.

[191] Hence, although Vermeulen discloses a "cloud-based" approach to determine available space. As explained in Claim 1[b] in Ground 1, the analysis required for storage is based on the limitation of a non-cloud-based filesystem as ext4, ntfs, or others where i-Nodes and other filesystem constraints require determining sufficient storage as, TotalFileSystemAvailable – Threshold. In Ex-1001 FIG 5, this is stated as follows:



[192] Therefore, Vermeulen does not disclose Element 1b by any means, as Vermeulen is a distributed system to determine space that can be applicable to Farber but not the ‘441 Patent Claim 1, Element 1[b]

[193] In the same analysis by Petitioner argues that Vermeulen can be combined with Farber. I disagree as undue experimentation is required:

a. First, Farber, each device has a storage as part of a personal computer,

b. Second, Farber relies on Data Sources 42, 44 to retrieve content, not to store it,

c. Third, Farber uses a triggering mechanism to coordinate all steps of encoding,

d. Therefore, a web-service systems with asynchronous behavior based on HTTP REST-style protocols. (Ex1005 at 5:7-15) “such as HTTP GET or PUT”, as explained by Vermulen (*Id.* 5:16-17) “Web services are stateless” whereas Farber is a state machine that follows a rigorous “triggering-driven” state-dependent solution.

[194] As the combination will Vermulen fails, then the analysis for Element 1[h] also fails, as again besides the conflict with Farber "Triggers" will need to be resolved, prior to combining Vermeulen "storage model for providing data storage to users of a service, such as a web service"

[195] Therefore Ground 2 for Claim 1 fails.

[196] Regarding Claim 2, the challenge to generate REST-style APIs and Farber is the fact that such combination will fail. As the HTTP Request/Response mechanism does not provide a way to "Trigger" an event, as disclosed by Vermeulen with HTTP GET, PUT, etc.

[197] In fact, the entire analysis with Vermeulen is flawed completely by Lipoff and Petition, 33 when it says that "Additionally, because a REST API, ***by definition, does not require a server to store information about a client*** (and client states and other parameters can be passed with the request) " HTTP by default requires a server such as Apache, Nginx, Python HTTP, and others in NodeJS. In fact, to store the state, as we should all know "cookies" are used to keep track of the client's state. That's why you are able to login back into your GMAIL account without entering your password all the time.

[198] As such, the '441 Patent teaches what is required o implement REST APIs, an HTTP Web Server (e.g. Nginx, Apache) ('Ex1001 at 8:30).

unit file system **730**. The RTSP server can be implemented using the Live555 RTSP server or similar. Alternatively, the HTTP Server can be implemented using "nginx" or "apache" web browsers, and the RTSP Servers can be implemented using Live555 services. In the case of the HTTP Server, this can also be used to get the Playlist of files generated from the method on FIG. **5 715**.

[199] Therefore, Ground 2 fails to disclose Claim 2

C. Ground 3 - Claim 3 - Regarding Faber and Pavlovkaia

[200] Regarding Claim 3, the language of the claim states that: "The method of claim 1, wherein creating the custom HTML user interface for each media file is performed using Javascript"

[201] Since this is a Method dependent on Claim 1, method is also not anticipated.

D. Ground 4, Claim 4, 5, and 8 are obvious in view of Farber and Davis

[202] Regarding Claim 4, with the language "The method of claim 1, wherein encoding each of the media files include encoding each of the media files at a specified bit rate."

[203] First for such combination, Davis needs to be able to handle "triggers" to replace the "Image Encoder" and the "MPEG Encoder" in Farber (Ex1002 at FIG 5 and FIG 4).

[204] The analysis by Petition, 37 is flawed, as considers the inexistent component "MPEG Video Encoder" as part of the combination. Stating that:

> POSITAs would have had a reasonable expectation of success in combining Farber with Davis's disclosure of a specified bit rate because Farber already contemplates use of an MPEG video encoder and an audio encoder and assembling media into streams for delivery and the specified rates of Davis could readily have been used to encode the audio and video data of Farber. Ex-1002, ¶146. Encoding data and selecting encoding bit rates were commonly-known to POSITAs and would not have required altering Farber's system. Ex-1002, ¶146.

[205] In fact, Lipoff at ¶146 state that Davis can replace the "Audio Encoder" and the "MPEG Video Encoder," both produce MPEG Transport Streams. However, Davis produces AAC audio with H.265 video codecs, as a consequence, undue experimentation to add Transport Streams will be required, as well as determining the functionality needed by the "Image Encoder" and whether all of these components need to be replaced. Therefore, Ex1002 confirms at ¶148 that Davis teaches "H.264 and AAC codecs" whereas Farber teaches "MPEG Transport Steams." An MPEG Transport Stream is an encapsulation or container format not a video encoding format.

[206] In fact at Ex1002 ¶149, Lipoff agrees with my prior statement and says that that "number of container format such as MPEG-TS and MP4 to deliver streaming

video," however when formats are exchanged, MP4 is not multiplexable by any means as MP4 is a storage format, an MPEG TS is a transmission format, that's why it is called MPEG Transport Stream.

[207] Once again, the analysis is flawed and therefore Ground 4 fails .

[208] Regarding Claim 5 and Claim 8

a. Claim 5 "The method of claim 1, wherein encoding each of the media files include encoding each of the media files using a H.264 encoder."

b. Claim 8 "The method of claim 1, wherein encoding the media file is performed using an AC-3 file format for audio content in the media file."

[209] Lipoff at ¶152 and ¶153 analysis is flawed by the reasons of Claim 4, as the Image Encoder, Audio Encoder, and MPEG Encoder in Farber will have to be modified with undue experimentation, including trigger synchronization mechanisms that are not disclosed by Davis.

[210] Therefor Ground 4 fails for Claims 4, 5, and 8.

E. Ground 5: Claims 6 and 9 in view of Farber and Fogel

[211] Regarding Claim 6 and the language "The method of claim 1, wherein the custom HTML user interfaces are configured to be rendered using a Webkit

browser" and Claim 9 language: "The method of claim 1, wherein the custom HTML user interfaces are configured so that multiple custom HTML user interfaces of the multimedia asset can be retrieved using a web rendering engine."

[212] Since Claim 1 is not satisfied, Claim 6 and Claim 9 is not met either.

[213] Simply for Fogel to operate, Farber will have to add a network card, a web-service engine, and a mechanism to use HTTP Protocol with the "Data Carousel." Such mechanism is not disclosed by Farber and in fact, such mechanism or an HTTP-based protocol that is "stateless" as Fogel describes will not be easy to integrate with Farber.

[214] Fogel at [0305] states that "based protocol, such as HTTP" that are stateless are incompatible with "trigger-based" machines that are state-dependent as explained for Vermeulen in Ground 2.

F. Ground 6 – Claims 10-16 and 18-23 in view of Avellan and Pavlovksiaia-PCT

[215] First, Claim 10 is an independent claim as follows:

[Pre] A computer-implemented method comprising:

a. virtualizing a system including a virtualized caching server, wherein the virtualized caching server performs:

b. 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;

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

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

e. 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;

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

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

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

a. Petition, 48 makes reference 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 Ex1002 ¶168 references that a "live streaming feed" such as a "Hockey Game" but "omits" that is a web page that contains a live streaming feed of a hockey game,

d. Avellan describes at *Id.* 13:41-46 that when web page "rendered by the remote browsers such "Hockey Game" 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, "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,)

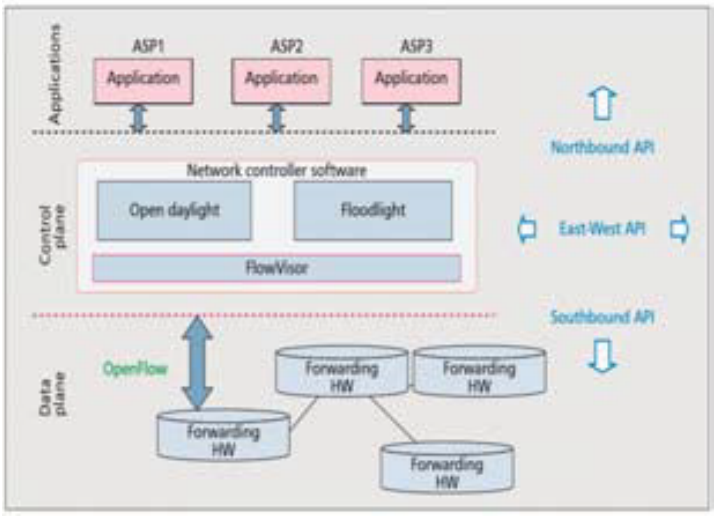
g. The browser renders image captures from a web page (*Id.* 5:8)

h. The "remote virtual browser" then "converts that to a form suitable for display on the – the devices 104, 106" (See Lipoff Deposition EX2053 60:1-3).

[217] In summary, I will start by showing that Avellan does not disclose Element 10[a], "virtualizing a system including a virtualized caching server, wherein the virtualized caching server performs"

[218] Element 10[a] is a "virtualization system of the caching server" and the methods of 10[b] to 10[h]. First, Pavlovskaia-PCT simply points to Figure 1 where the "Virtual Machine" is referenced in the patent description. However, there is no disclosure that such the VM cited performs the methods for Claim Elements 10[b] to 10[h]. Indeed, just referencing a "Virtual Machine" word, oversimplifies complexities at virtual machines (Ex2050)[4], as shown in the figure below, Software Defined Networks are usually obtained as "abstractions" by the Virtual Machine. The most common ones are defined by "Network Controller Software," in Java and Android an abstraction called "Radio Interface Layer" permits, Java applications use phone recourses such as 4G and 5G interfaces. In other systems, these interfaces are GbE, 10GE, and even multi-Gigabit interfaces, that are slices and managed accordingly.

[4] A. Rista, J. Ajdari, X. Zemuni. "Cloud Computing Virtualization: Comprehensive Survey," MIPRO2020, October 2nd, 2020 – IEEE Explorer.



[219] Pavlovkaia-PCT only has one "virtual machine" at 106,

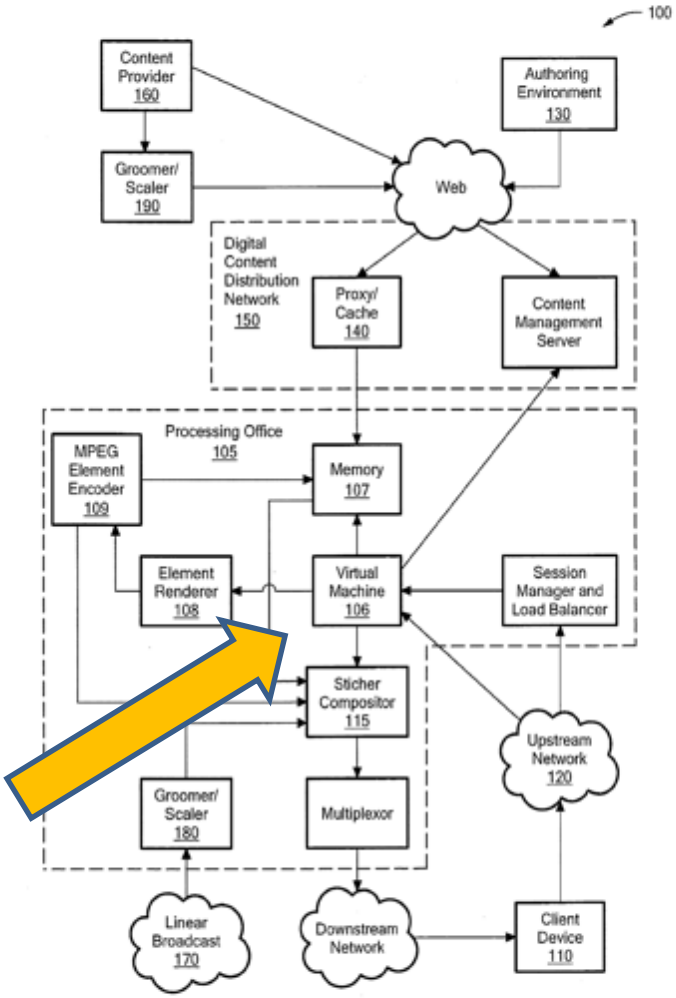


FIG. 1

[220] The "orange arrow" points to the only "Virtual Machine" 106 in Ex1007 at FIG. 1. Therefore, by no means Pavlovkia-PCT suggest the "virtualization" of the Claims 1[b] to [1g]. In fact, the combination with Avellan is just not possible, even if Avellan needs to be virtualized, it can't, as the satellite system cannot be

virtualized. Some LAN to Satellite system will have to be used to create a "Virtual Satellite Interface" that can interact with a potentially virtualized system that includes a "Broadcast interface." Besides, understanding on how the "Web-to-HD Video Protocol" is crucial to determine if, given its protocol status, whether it can be virtualized or requires an interface for it. Therefore, undue experimentation would be required to virtualized a "satellite-based" broadcasting platform.

[221] The '441 Patent discloses in great detail as FIG 10 and FIG 11, and *Id.* 12:5-41, how virtualization is done a) Interfaces and Modules into Virtual Machines or Docker Images, and b) Method for provisioning, deploying, and instantiating those VMs.

[222] Regarding 10[b] that reads "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;

a. First, Lipoff declared that the "Red Arrow" at Ex1002 ¶93 points to the "Content Provider" or EX2054 the "Blue Box" at his deposition.

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,

[223] Avellan does not disclose a "one media stream for playback", as the "one media stream" is defined by the '441 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[5], HTTP/RTSP, RTSP[6] Stream) *Id*. 8:14-16 or "HTTP Playlist" *Id.* 7: 47-51

[224] 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)

[225] **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. Therefore, Avellan teaches "compresses the image or video

[5] EX2056 – HTTP Live Streaming Protocol Specification

[6] EX2048 – RSTSP Streaming Protocol Specification

frame format and sends it to the browse 106", also cites the use of "Compressed video file" and a "Tag file" at *Id.* 5:44 and "displayed (once decompressed)" at *Id.* 7:47

all the information on the web page even if that information continues over a period of time. Once the page is imaged (including any audio on the web page), the gateway server **134** compresses the image or video frame format and sends it to the browser **106** via the satellite **102**.

various locations of the action points on the web page. That tag file is associated with the compressed video file based on the domain name for the imaged page, and the video is syn-

the broadcast device **102** to the user computer **104**, where it may be immediately displayed (once decompressed at the computer **104**). Accordingly, the unicast mode provides fast processing which is achieved using a fast differential burst

[226] 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.

[227] Hence, Avellan does disclose the "at least, one media streams for playback" due to each web page bring stored as compressed video that is also with "tag files" that in Avellan contain positioning information of all items on the web page. Hence,

the compressed video are treated as files to match the TAG File used for interactive clicking over items, for example:

a. As shown the traffic at FIG 1, element 120, 122, 124 are: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.

[228] Moreover, Element 10[c] 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"

[229] 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.

[230] 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 are not going to provide a good user experience.

[231] For Element 10[c], "obtaining content corresponding to the plurality of multimedia items from at least one source" implies that there could be multiple sources, not only one. The '441 patent discloses this process as follows:

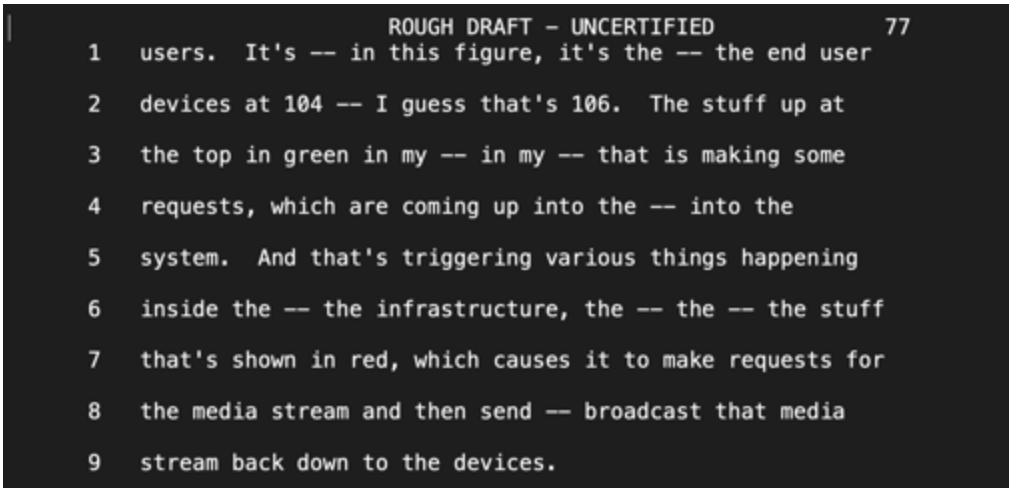
a. Via "rsync" protocol, See Ex1001 at 4:13

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.

[232] Hence, the only mechanism where the Petition and Lipoff points to is by a request made by the Gateway server to the web-page.

[233] In fact, Lipoff repetitively points to the webpage to be a "media stream." As in 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 Deposition at EX2053 Pg. 77:1-9

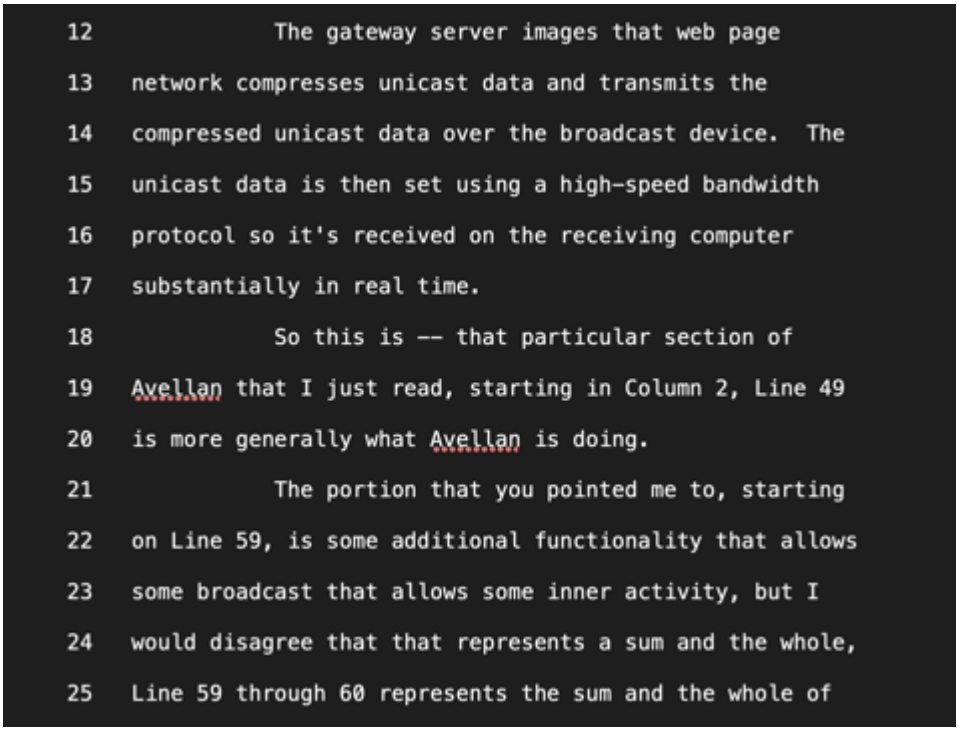
[234] Obviously, a "web page" is not a "media stream," hence the Claim 10[b] and [c] deals with multimedia content other than the web page.

[235] 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 (EX2053) at 130:5-6.

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

[237] Now For Element 10[d] 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 EX2053 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
```

Lipoff Deposition (Ex2043) at Pg. 66: Lines 12-25

[238] Hence, the Petition and Lipoff relies on the “Web-to-HD video protocol” presented by Avellan or te “Web to HD Video Protocol” is referenced by EX1004 at 5:16-32 and briefly mentioned by Lipoff at ¶124 in EX1007 at IPR2025-00350.

[239] There is not a single publication online or in the Petition files that describes how “Web to HD Video Protocol” works.

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

> 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

[241] 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" Instead, the '441 patent teachings do not require of any "video protocol" or communication to achieve the functionality disclosed.

[242] Instead, the '441 provides a detail way to convert web pages into images and then to media streams as follows:

a. By using PhamtonJS 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 '441 patent.

d. Next, the captured images for the "Custom User Interface" are used as inputs for the "FFMPEG"[7] 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[8] with other multimedia files (e.g. Music), that were not part of the "Custom UI" process (See *Id.* at 7:39-40)

f. The '441 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),

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

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

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

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

a. Compresses web page images captured into a video (as video snippets from the web page),

b. User computers retrieve these compressed files, that required decompressing them for displaying

[244] Therefore, Element 10[f] 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," as that rationale is simply flawed.

[245] Regarding Element 10[g] 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 '441 patent, 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”

[246] Additionally, Petition, 51 addresses Claims 11a as disclosed by Avellan, I will show this statement is false.

[247] I will start my analysis with Claim 11a’s language “The method of claim 10, wherein the obtaining comprises: retrieving, for each of the plurality of multimedia items, at least one audio file corresponding to an audio component of the multimedia item and a plurality of screen captures corresponding to a video component of the multimedia item “

[248] Petition fails to show that Claim 11a is present for the following reasons:

a. Claim 11a, that depends on Claim 10, that is not disclosed,

b. However, Claim 11a requires that "retrieving, for each of the plurality of multimedia items, at least one audio file corresponding to an audio component of the multimedia item" hence, within the process of obtaining multimedia items, one of the items can be an MP3 file. Petitioner points to the “the page is imaged (including audio on the web page) at Petition Pg. 51-52," however this is not part of obtaining “multimedia items” instead part of “Rendering a Web Page”

c. And, the rest of Claim 11a states that “and a plurality of screen captures corresponding to a video component of the multimedia item."

d. Clearly, the screen captures, and the video component are one separate element from the audio file, and in Avellan, the audio file is part of the webpage and the screen captures generated as a “compressed video file”.

[249] Additionally, Petition, 51 states that

> “Once the broadcast data is stored in the gateway cache 136, the gateway server 134 then transmits that data to the multiple user computers 104.” Ex-1004 at 8:54-56 and 8:18-21 (explaining that system 100 can be used to broadcast “live streaming television feeds” and “movies”). As a result, the gateway server 134

[250] Since the audio of a live feed is not in a separate file, nor the a video file contains a separate audio file. The evidence presented by the Petition and Lipoff fail to show at least one audio file under these terms.

[251] Additionally, even if the audio file is loaded as part of the web page, image capturing into a video. The audio file would be retrieved and compressed within the video, and it will require experimentation and testing to retrieve an audio file embedded on a webpage, especially if a web-player is required.

[252] Hence, Element 11a fails under Ground 6.,

[253] Regarding Ground 6 and Claim 12a, that also depends on Claim 10. Petitioner fails to show Claim 12a as well:

a. First, Claim 12a requires that "wherein the generating comprises: combining the plurality of screen captures and the at least one audio file to create each of the plurality of multimedia items,"

b. The language of this claim requires that "combining the plurality of screen captures and the at least one audio file",

c. This indicates that "screen captures" and the "at least one audio file" are separate. In Avellan, when the screen captures take place, if an audio file is involved, the audio file is captured as well,

d. Hence, the claim requires an additional step "to create each of plurality of multimedia items"

e. Since Avellan, can only get screen captures at most, the rest of Claim 12a are not met,

[254] For Claim 12a, Petition, 52 points to a "live stream " at EX-1002 ¶112, and "live streaming feeds" and "movies." However, live streams and movies do not have

a “Separate” audio file, simply a) A live streaming feed has an audio “Program Identifier,” not a separate audio file, and b) Movies also have an audio-track not a separate “Audio file.”

[255] All users of a streaming service know that we watch a live stream in a web page, there is no separate “audio file” downloaded. Simply, having a separate audio file is incompatible with a live feed, as synchronization is hard to impossible.

[256] Regarding Claims 13a to 13c that depends on Claim 12, with the following language:

a. “13a. The method of claim 12, further comprising: the virtualized caching server detecting a change at the at least one webpage corresponding to at least one of the plurality of multimedia items

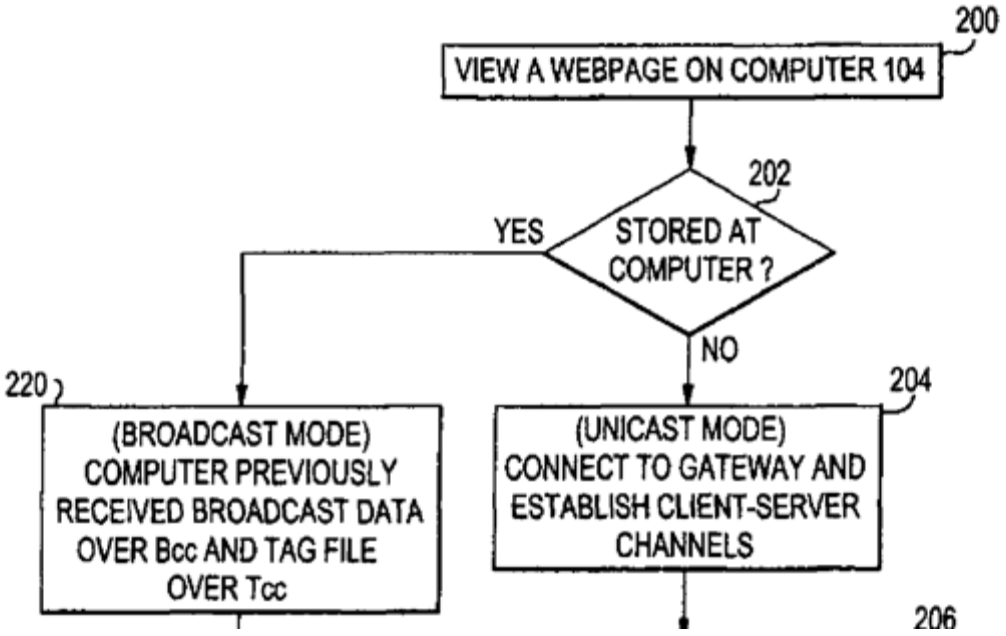
b. 13b. in response to detecting the change, retrieving a new plurality of screen captures from the at least one webpage for the at least one of the plurality of multimedia items; and

c. 13c. reassembling the at least one of the plurality of multimedia items using the new plurality of screen captures to yield an updated multimedia item.

[257] The Petition, 53 points to the "the gateway server 134 can continuously or periodically compare" such comparison for a change has to be done within "compressed files," it is unclear how Avellan can make this comparison, as Avellan simply checks for domain names, not for the content of the file on itself.

[258] Element 13a and 13b involve from Claim 10 the "of the plurality of multimedia items," since Avellan failed to show obtaining this plurality of multimedia items, then Avellan to disclose Claim 13.

[259] Additionally, the method suggested by petitioner and disclosed with Avellan is shown in Ex1004 at FIG 2. and uses domain names See *Id.* 5:45



[260] Hence, Avellan compares the "Broadcast Data" and updates the information, as the information is updated, the "**updated data is compressed and**, saved to cache

136". Once again, it further confirms that "compressed data is stored in the cahe, and the "Storage device 110"

tinuously or periodically compare the broadcast data stored in the cache **136** with the broadcast data at the networks **150**, **152** and re-capture any broadcast data which is updated. As the information is updated, the updated broadcast data is compressed, saved to cache **136**, broadcast to the user computers **104**, and stored in the storage device **110**.

Ex1004, 9:1-6

[261] In terms of Claim 14a, with language "The method of claim 11, wherein the retrieving of the plurality of screen captures comprises obtaining the plurality of screen captures from a playback of a video on the at least one webpage" also depends on Claim 11. The Petition at Pg .54 addresses this, and states that:

> Avellan teaches that "web pages often have information which is rendered over a period of time, ***such as a video***, streaming information, or an animated display. Accordingly, the gateway server 134 ***continuously captures images of the web page*** until it detects a repeating pattern of the information in the web page." Ex-1004 at 5:5-5:10. And Avellan explicitly states that "suitable sites for

[262] Again, since Claim 11 is not disclosed, Claim 14a is not disclosed.

[263] Claim 15a which states that “The method of claim 14, wherein the combining of the plurality of screen captures comprises metadata corresponding to at least one of a song title, an artist, and a music genre”. The Petition, 54 points to the “Country Music” reference at Ex1004 8:25-27 and Ex1002 ¶173.

[264] I disagree, as the metadata with song title, artist, and music genre, as it depends on Claim 15a, neither Claim 14 nor Claim 11 are met, hence this is not met.

[265] Claim 16a which states that “The method of claim 10, further comprising: the virtualized server identifying the at least one webpage based on a channel identifier associated with the broadcast media channel.” Contrary to the Petition, 55, that claims the “Channel Identifier” used to “the user to tune to a particular channel.”

[266] This is not the case for the ‘441 Patent, setting a channel identifier for a “user to tune to a particular channel” is done by the “operator,” instead the channel identifier is used to map multicast addresses in the broadcast channel as a tuple (Ex1001, 10:13-15), for example, or as an identifier to passed to the HTTP web service that retrieves, an HTTP Playlist, and HTTP Live Streaming feed (Ex1001 7:11-15).

[267] Therefor Ground 6 fails for Element 16a.

[268] Claim 18a states “The method of claim 10, wherein the content provider is a cable television operator or a satellite television operator“, for this claim Petition, 56 switches include Element 102 as part of the “Content Provider”

[269] As Claim 10 is not met by Petitioner, then Claim 18a is not disclosed by Avellan.

[270] Element 19a which states “The method of claim 10, wherein providing the at least one media stream to the content provider comprises streaming the at least one media stream to an MPEG multiplexer associated with the content provider.”

[271] For this claim Petitioner relies on Pavlovskaia-PCT to bring the “MPEG Multiplexer.” However, Avellan discloses the use of “compressed data” Ex1004, 9:1-6 and cannot use of an “MPEG Transport Stream” that is compatible with a Multiplexer.

[272] Petition, 57 cites Ex1004 5:13-15, as shown herein:

> continues over a period of time. Once the page is imaged (including any audio on the web page), the gateway server **134** compresses the image or video frame format and sends it to the browser **106** via the satellite **102**. 15
> In a non-limiting aspect of the invention, a web-to-HD

[273] By no means, Avellan discuss the use of MPEG or suggests its use, as simply operates the gateway sever such that “compresses the image or video frame format and sends it to the browser 106 via satellite 102”

[274] Therefore, such combination will fail and will require undue experimentation to determine how the Multiplexer will work and whether such Multiplexer will be placed inside the satellite or, if will handle different multicast channels. As Avellan already works with multicast groups and cache distribution, as shown at Ex1004 12:14-23.

feature of the invention is to provide one or more unidirectional Multi-cast Communication Channels (Mcc), each of which carries transmissions to a predetermined a group or subset of users. For example, users may decide to join a multi-cast group, and communications unique to that group would be transmitted to those user computers **104**$_1$-**104**$_n$ over that multi-cast channel. For example, the caching engine at the gateway server **134** may push the cache **136** periodically onto the computers **104** by multi-casting it to subscribers. In the example above, the user computers **104**$_1$-**104**$_n$ located in Florida can be assigned to a common multi-cast group having a designated multi-cast channel. By storing a portion or all of

Ex1004, 12:14-23

[275] Undue experimentation will be required to coordinate not only changes to the "push the cache 136 periodically onto the computers 104 by multi-casting it to subscribers" but also changes to the remote browsers, as changes need to be made, as clients in Povlavkaia-PCT (Ex1007 3:7-11) are not compatible with the Avellan users at 104 (See Ex1004 at FIG1). Both clients work with interactive content on different formats, First, Avellan uses TAG files and compressed data with computers as users that count with storage devices 110 (See *Id.* FIG 1), while Povlavkaia-PCT is designed for client devices "having an associated display, such as a Television"

Ex1004: 6:15-16; Second, Avellan connects with web pages, while Povlavkaia-PCT uses AVML which is an XML-based file system defining MPEG slices (See Ex1007 7:3-5) Among many other problems, that include Povlavskaia-PCT multiplexer from Povlavskaia-PCT handles data and video, hence it will be unclear how the multiplexer will be used with Avellan. Ex1007 24:7-9)

[276] Hence, Element 19a is not disclosed by Ground 6.

[277] Element 20a states that "The method of claim 10, further comprising: the virtualized caching server provisioning, on at least one server, at least one virtual machine for performing the receiving, obtaining, generating, assembling, and providing." Provisioning is described by the '441 Patent in great detail in FIG.11 and in the specification *Id.* 12:15-30, 12:44-57 and 13:1-45

[278] The Petition, 61, does not address that "virtualized caching server provisioning, on at least one server" Even Ex1002 ¶129 assumes that "orchestrated by a virtual machine" suffice. This is incorrect and falls short on what Virtual Machine provisioning requires.

[279] The '441 patent teaches how provisioning is done of a Virtual Machine. See Ex1001 at FIG 10 and FIG 11, and the narrative on *Id.* 12:15-30, 12:44-57 and 13:1-45.

[280] Again, the virtualization of different network ports, storage, provisioning of IP Addresses, credentials, etc, is not disclosed by Pavlovskaia-PCT and, in general since the "satellite 102" at Ex1004 FIG1 is part of the "content provider" such provision and virtualization is simply not possible.

[281] Element 21a " The method of claim 10, wherein virtualizing the system is performed using one of Docker, Xenserver, or a virtual machine," Petition, 60, states an oversimplified incorrect reasoning, when the '441 patent teaches how provisioning is done of a Virtual Machine. See Ex1001 at FIG 10 and FIG 11, and the narrative on Id. 12:15-30, 12:44-57 and 13:1-45. Hence, a POSITA will use the disclose on FIG 10 and 11 to create a virtual machine with Docker or XenServer, as well as the method of Claim 10. As observed, from EX2050 Figure 6, the physical interfaces are abstracted by using bridges, VLANs, and exposing those to the virtual machine environment (e.g VMWare, Xen, Docker), or even a "Java Virtual Machine" or "Dalvik-VM" (EX2060, 8-9, 28 Table 4.1) exposes APIs to "abstract" the hardware

Figure 6. Network elements within an OpenStack network node [13].

[282] Element 22a "The method of claim 10, wherein virtualizing the system includes virtualizing the system to include at least one virtualized multicasting server that perform the providing at least one media stream to the content provider." This is also not debatable and incorrect analysis given by Petition, 60-61, as shown in Element 21a. The virtualization can also be done fur the multicasting server, as well as the system, such that copies of the same virtual machine disclosed in FIGs.10 and 11, are used to provision, and deploy different channels with different cloud instances.

[283] Element 23a "The method of claim 10, wherein virtualizing the system is performed for one channel, wherein the virtualized caching server is operable for one channel, wherein virtualizing the system is repeated for each one of a plurality

of other channels." For the same reasons disclosed for Element 21a and 22a, where the provisioning and the virtual machine disclosed in FIGs 10 and 11 together with Claim 10 can be deployed into a cloud to services a plurality of channels. The literature indicates that elements such as Software Network Interfaces or Software Defined Networks are key for virtualization (e.g. EX2050, pg 462, 2nd column and Pg. 463 first column, Figure 1)

[284] For all these reasons, Ground 6 of the Petition fails and all claims associated to this ground are valid.

G. Ground 7: Claim 15 is obvious in view of Avellan, Pavlovskaia-PCT and Durante

[285] Claim 15a states that "The method of claim 14, wherein the combining of the plurality of screen captures comprises metadata corresponding to at least one of a song title, an artist, and a music genre."

[286] Moreover, Lipoff analysis on Ground 7 and Durante fails for the following reasons:

a. Durante does not disclose "Metadata" being available thru an API, to be inserted as part of the "one media stream," but instead discloses a video library (See Ex-1004 at 4:19-21) with its

internal metadata. As the metadata is already part of a video library, there is no need to expose it via an API

b. This metadata is part of the web-page rendering mechanism attributed to "Avellan's Web-to-HD Video Protocol."

c. Hence, Durante is not combinable with Avellan, as Avellan requires the video to be part of the "web-page", and Durante's video library will have to displayable via the "web page" in a way that Avellan can use two web pages, which simply can't, as the HTTP protocol requires all web-pages to be within the same domain

d. Therefore, undue experimentation will be needed to:

e. Comvert Durante's Video library and process each video to be converted to audio-only, b) Convert or extract the text in the video to be part of the web-page to be rendered by Avellan, and c) If the web-page needs a "web player" such web-player may assume full interactivity with the screen.

f. Integration with Avellan's Broadcasting system simply can't work, because if end-users will interact with the video library from Durante has to be done in the Unicast channel. For a

broadcast, a plurality of multimedia item from such library will require to be either individually selected, and the creation of playlists to be automated and played within the web-page.

H. Ground 8 - Claim 16 in view of Avellan, Pavlovskaia-PCT, and Ma

[287] Regarding or Ground 8 for Claim 16a "The method of claim 10, further comprising the virtualized caching server identifying the at least one webpage based on a channel identifier associated with the broadcast media channel."

[288] Lipoff at ¶125 states that "Avellan converts web content into a compressed "video frame format" that is "suitable for broadcasting to a large number of user computers," however this is insufficient for Claim 10, which means that Ma in combination with Avellan will face a challenge as the following mapping collides:

a. Ma purports to assign a channel ID to yahoo.com

b. Avellan compresses, assume Ma assigned #1 to yahoo: "file1" for "user1," "scrolls to the top," "file1" arrives for "user1",

"scrolls to the bottom," then "user1" will see his commands affected by user2.

c. Therefore, such assignment requires, not only an identification of each channel portion, but also each user requesting it, as each user may interact with each channel differently, via the TAG File.

d. The main issue is that Ma identifies Channel that are static, and have no interactivity, whereas Avellan provides tag files and interaction.

[289] The combination with Ma and Avellan requires undue experimentation and this rationale fail for Ground 8

I. Ground 9 - Claim 17 in view of Avellan, Pavlovskaia-PCT, and Suzuki

[290] , Regarding Ground 9 the Claim 17a recites that "The method of claim 10, further comprising: the virtualized server querying the content provider to determine the second format, wherein the second format corresponds to a Moving Picture Experts Group (MPEG) format.

[291] The combination of Avellan with Suzuki enters into a contradiction, a) Avellan relays on "output "video frame format" was able to be decompressed an display at devices." Avellan extracts the TAG File and the "video" from its format

and provide captured images to display, b) MPEG encodes and decodes content, hence if Avellan as a "Content Provider" uses "Suzuki" to choose a "decoding format" (Ex1012 at 3:44-49). Hence, Avellan will have to change how a) Tag Files are delivered to the user, b) Instead of decompressing files the "Web to HD Protocol" will need to operate with encoders and decoders.

[292] However, the roadblock is that the "Web to HD Video Protocol" specification is an unknown. However, this imposes a problem for Avellan's gateway as how to deal with audio files as well.

[293] Inconsistent mapping, as shown by EX2054, the "users' computer is only unit capable or acting as in "Suzuki" that contradicts the mapping, as the query is am outgoing request, not an incoming request. Therefore, Ground 9 fails as well.

1. Ground 10 – Claim 24 is obvious in view of Avellan, Pavlovskaia-PCT and Gangadharan.

[294] As shown at, Claim 24a, recites "The method of claim 10, wherein receiving the request is performed using a JSON API."

[295] However, Ground 10 argument by petitioner switches the content provider from being "Red Arrow" to be the receiver Laptop on Avellan's FIG 1 Element 104 and 106.

[296] Additionally, Lipoff at ¶¶145-146 switches his position from the "content provider" being the "Gateway" to be the Laptop and Petition, 53 is shown herein:

> Similarly, Avellan states that the user computers 104 "can be any suitable device . . . such as a laptop computer, desktop computer, smartphone, tablet PC, laptop, and Internet equipped television," which a POSITA would recognize would all be capable of communicating using Internet Protocols. Ex-1003 at 3:53-56 and

[297] First, Petition fails to show a well-defined request as in Claim 10. Avellan requests a URLs from the "users computers at 104" Ex1007 at FIG.1. Petition is now saying that this JSON Requests are formed from the users computers to the gateway, hence the gateway needs to have a way to convert POST, GET request, and parameters in JSON format back to standard HTTP Requests for the web sites (e.g. Yahoo), as yahoo.com expect a standard HTTP Request.

[298] However, inconsistent analysis by Lipoff where EX2054 "in blue" is the "Content provider" switches to be the "user's computers, not the Gateway as established before. Therefore, Ground 10 also fails.

2. Ground 11 – Claim 25 is obvious in view of Avellan and Wannamaker.

[299] The language of Claim 25 includes 25[pre] A computer-implemented method comprising:

[300] Claim 25[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;

[301] Claim 25[b] obtaining content corresponding to the plurality of multimedia items from at least one source offering the content in at least one first format, performed over a first TCP-IP socket of a caching unit;

[302] Claim 25[c] rendering a web page by a browser using the content;

[303] Claim 25[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;

[304] Claim 25[e] assembling the at least one media stream using the temporal sequence of screen captures

[305] Claim 25[f] providing the at least one media stream to the content provider for broadcast on the broadcast media channel; and

[306] Claim 25[g] "providing, at a second TCP/IP socket of the caching unit, a status of the obtaining, generating, and assembling."

[307] Petition, 69 states that "For the same reasons on Ground 6 for 10[pre] and 10[b] to 10[g]" which I stablished previously that Avellan fails to disclose, then asserts that Avellan discloses elements 25[a] to 25[f].

[308] In summary, Petitioner still assume that a media stream is a "web page" and the "compressed video files" provided by Avellan constitute "media streams,"

[309] However for Element 25[b] and the term "performed over a first TCP/IP socket," the passement made that Avellan by virtue of the HTTP Request made to the "Internet Cloud" classifies as performing this functions.

[310] This statement is incorrect, as the use of the TCP/IP socket is to start the process of "obtaining content corresponding to the plurality of multimedia items." The '441 Patent describes that besides Port 80 that corresponds to HTTP, a port, for example, 9553 is used to start or initiate the "obtaining content" step. (See *Id.* 9:31-40)

> In some embodiments, there are two possible TCP/IP sockets open for listening used by the Caching component of the Mediaplug. A) Port 80 (as an example), or the standard HTTP port, used to retrieve content, view video content generated, update/delete cached files, and access the play-lists, and B) Port 9553 (as an example), or the standard Mediamplify Parallel encoder port, used to monitor and review status of the songs being encoded, the data being collected, and all other related activities with encoding. Via this port, encoding/transcoding can be started or stopped.

‘441 at 9:31-40

[311] As shown, the ‘441 teaches the use of incoming ports 80 and 9553, which are used to “retrieve content, vide video content generated, update/delete cached files, and access the playlist”, and port 9553, used to monitor, review status of the songs being encoded, and … via this port, encoding/transcoding can be started and stopped”

[312] In general, a POSITA will know that in order to “obtain a plurality of multimedia items” a TCP port is used to “connect to,” e.g. retrieving content via RSYNC using SSH port 22. However, the use of a TCP/IP socket is used to be an incoming port to initiate “obtaining” as depicted by Claim 25[g] where the claim language specifies “providing, at a second TCP/IP socket of the caching unit, a status of the obtaining, generating, and assembling.”

[313] Precisely, only an incoming port can provide information or status for obtaining, generating, and assembling, as for example a request to “getStatus” information will provide a reply message as shown at Id. at 9:20-30.

[314] Therefore, Avellan discloses a port at the Internet cloud that receives request, by no means Avellan discloses any “incoming ports” in used to initiate “obtaining”

[315] A POSITA cannot combine Wannamaker and Avellan as expect that OMI, to be used for “obtaining.”

[316] Obviously, the Petition and Lipoff at Ex1002 ¶204 confused the nature of the ports in 25b and 25f, hence their combination analysis is flawed.

a. First, Wanamaker at Ex1006 [0087] provides a boiler plate statement on generic use of "logs", shutting down a system, had to disclose processing of the request and tracking retrieval status of each of the multimedia item.

b. Second, Wannamaker does not provide any evidence of "obtaining, generating, and assembling."

c. Third, without giving explanation Lipoff at ¶203,204,205 relies on how a POSITA "will know" and errs at ¶205 as the sockets described by Avellan are "outgoing sockets" not incoming sockets.

[317] Therefore, Ground 11 fails to show that Claim 25 is anticipated.

3. Ground 12 – Claim 26 is anticipated by Avellan

[318] Element 26a to 26e are the same as described in Ground 11.

[319] Element 26a, A computer-implemented method comprising at a caching unit:

[320] Element 26b, 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;

[321] Element 26c, obtaining content corresponding to the plurality of multimedia items from at least one source offering the content in at least one first format;

[322] Element 26d, rendering a web page by a browser using the content;

[323] Element 26d, 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;

[324] For this analysis, Petitioner then switches to "computer and laptop 104" to be "Content Provider", as per Lioff Analysis at ¶208, where he states that:

> 208. With respect to Element 26h, Avellan teaches that "preferably more than one server 134 stores Internet data in a cache 136 ***for redundancy*** and to better distribute communication channels for that server 134." Ex-1004 at 9:55-57. POSITAs would recognize that the concept of "redundancy" in servers means that Avellan therefore teaches that when a fault is detected at a caching unit, pre-recorded video images can be retrieved from memory associated with a different caching unit.

[325] Although redundancy provides more servers, it does not disclose using a "recoded media stream," which is related a failure such as: loosing network access (Rendering a Web Page has failed), errors in obtained media (e.g. Assembling step failure), in any of these cases, if the same media is being used for "Redundancy" the fault will be replicated in all servers, and all servers will fail. Therefore, the novel

solution prosed by Claim 26, uses a "recorded media stream" from a previously operational system that can clearly be used from "one media stream"

[326] Petition, 72-73 does not rely on any Lipoff analysis except at Pg. 73 for Ex1002 ¶207 and ¶¶210-211. In this analysis, Petition cites Avellan at Ex1004 12:11-18 and says that because "users may decide to join a multicast group" a multicast unit exist, as required by claim 26f language that recites:

"at a multicasting unit that is operable connected to the caching unit: providing the at least one media stream to the content provider for broadcast on the broadcast media channel;"

[327] However, and as stated before,

a. Avellan relies on compressed video files that are sent to the "Broadcast Device" at *Id.* element 102 of FIG 1,

b. Avellan does not disclose a "one media stream" instead Avellan operates in compressed video files that required "may be immediately displayed (once decompressed at the computer 104" at *Id.* 7:56-48

c. Avellan can operate in Burst mode for very short periods of time, and a more "uniform lower-bandwidth transmission over a longer period of time" See Ex1004 7:16-20.

[328] Element 26f, “at a multicasting unit that is operable connected to the caching unit: providing the at least one media stream to the content provider for broadcast on the broadcast media channel;”

[329] This Element is not disclosed either by Avellan, as a multicasting unit requires of a “Playlist” to generate a “one media stream” and a “multicast address”

a. Avellan does not provide a playlist, simply retrieves “compressed video files” that require to be decompressed to be displayed (See Ex1003 at 7:57)

b. Avellan does not disclose a “providing the at least one media stream” as “compressed video files” are not media streams.

c. Additionally, Avellan will have to show a mapping of media streams, not web pages to a set of “multicast addresses” (See Ex1001 at 10:13-15

[330] Now for Element 26g “recording the media stream from the caching unit to produce a recorded media stream; and”

[331] This step is for “recording the media stream” that is being broadcasted, as that media stream already has been processed and includes the “web page” with all rendered resources, and the “plurality of multimedia items” which is then tested by the user. Petition, 72 points to the “caching unit at the gateway,” hence switching

the content provider to be the laptop 104 not, what Lipoff testified at "Exhibit #3." Even under this analysis, the web-page image captures are the ones stored at the "caching unit," as Avellan does not disclose a "one media stream for playback," Petition fails in their analysis.

[332] Last, Element 26h "upon a fault being detected at the caching unit, providing at least a portion of the recorded media stream to the content provider." Again, Petition, 73 points to "Redundancy in Avellan"

[333] The evidence cited by Petition, 73 and 74 and Lipoff Ex1002 ¶210, as the fault is detected in the "Caching unit," not in the system, not in any component. The analysis by Petitioner fails for several reasons:

a. Lipoff points at the Content provider as "the Red Arrow" at Ex1002 ¶93 or Exhibit #3 of Lipoff Deposition.

b. In contrast, Lipoff Declaration, Ex1002 at ¶208 appears to state that the "caching unit" is the "content provider" or the Gateway server at Ex1004 9:55-57 as "cache 136 ***for redundancy***"

[334] However, assuming switching appropriate, redundancy will not work if the fault in the caching unit represents that a failure in the encoding process, transcoding, retrieving, or other failures have occurred derived from a media file that is faulty, or the web-page providing a 404, error. In any of these cases, "redundancy" simply

will multiply the error by a number of servers. Hence, the solution using a "recording" is better than redundancy and more appropriate for multimedia systems.

[335] For these reasons, Claim 26 is not anticipated by Ground 12.

4. Ground 13 – Claim 26 is obvious in view of Avellan and Wannamaker.

[336] Ground 13 only covers Element 26h which reads "upon a fault being detected at the caching unit, providing at least a portion of the recorded media stream to the content provider" and points to Wannamaker Ex1013 at [0101].

[337] Again, rerouting transmission will be also useless, as the media is the cause of the fault.

> application server, forming one logical server." Ex-1013 at [0101]. "If a server fails, transactions are routed to another server in the server group." Ex-1013 at [0101]. Wannamaker also teaches that each server "caches the converted content for future reuse." Ex-1013 at [0270]. "The rendered 36 cache is a storage area for

[338] Clearly, the converted content is cached for "future use" means that is a "Cache" in other words, if it was already "hit" then will be standard "Cache miss" and "Cache found" mechanism (See Ex-1013 at [0270]), This is similar to FIG 2 in Avellan, when checks if the web-site is stored in cache and does not need to recover it. However, the failures in multimedia systems include, having a valid request, but

the media being invalid due to different "Codec," "Corrupted media," "DRM" issues, and other failures related to multimedia systems.

[339] Therefore, the analysis made by Lipoff at Ex1002 ¶¶209 and 211 is done from the perspective of standard routing and server failures, not due to failures with relation to media failures that can only be compensated with a valid prior-recording tone for a an operational system.

[340] In fact, I completely disagree, as Avellan's operation will require substantial modifications from Wannamaker. Wannamaker works with a "microbrowser" Ex1013 at [00267-273] that works with MML pages and some MPEG I-frames. The sessions started by the microbrowser are substantially different to Avellan, as Avellan uses a "virtual browser" and "browser **106"** at Ex1004 at FIG.1 Element 104 and 7:1-2, 8:2 , this is quite the opposite to Wannamaker MML pages and MPEG I-frames. Simply, Avellan's browser and Wannamaker microbrowser are not compatible and cannot be combined without undue experimentation.

[341] For these reasons Claim 26 is not anticipated by Ground 13.

IX. CONCLUSION

[342] Grounds 1 to 26 fail to show that any of the claims of the '441 patent is invalid user U.S.C. 102 and 103 grounds.

[343] 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.

X. CERTIFICATION

[344] 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 12, 2025 Respectfully Submitted,

Edwin A. Hernandez. Ph.D.