

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AYLO FREESITES LTD.,
Petitioner,

v.

WELLCOMEMAT, LLC,
Patent Owner.

Case IPR2024-00710
Patent 8,307,286

EXHIBIT 2015

DECLARATION OF MARK P. MAHON, Ph.D.

TABLE OF CONTENTS

	<u>Page</u>
I. INTRODUCTION	1
II. QUALIFICATIONS AND PROFESSIONAL EXPERIENCE	6
III. RELEVANT LEGAL STANDARDS	14
A. Burden of Proof	14
B. Anticipation (35 U.S.C. § 102).....	14
C. Obviousness (35 U.S.C. § 103).....	15
IV. LEVEL OF ORDINARY SKILL IN THE ART AND PRIORITY DATE	19
V. THE '286 PATENT	20
A. Procedural History	20
B. Overview of the '286 Patent	21
C. Prosecution History	28
D. The Claimed Invention	30
VI. CLAIM CONSTRUCTION	31
VII. SUMMARY OF THE PRINCIPAL APPLIED REFERENCES.....	32
A. Gupta (Ex. 1004).....	32
B. Fegan (Ex. 1005)	33
C. Kim (Ex. 1006)	33
D. Vasilevsky (Ex. 1008)	34
E. Logan (Ex. 1009).....	34
VIII. CLAIMS 13-15 AND 17-20 ARE NOT ANTICIPATED BY GUPTA (GROUND 1).....	34

A.	Claims 13-15 and 17-20 Are Patentable Over Gupta	34
B.	Gupta Fails to Anticipate Claim 13 of the '286 Patent.....	35
C.	The Ordinary and Customary Meaning of Limitation 13.1 In View of the Intrinsic Evidence	35
1.	Prosecution History	36
2.	Claim Language	38
3.	Disclosure in Specification of the '286 Patent	39
D.	Gupta Fails to Disclose <i>Providing a Computer System ... Comprising a Video Editing Environment, and Being Disposed at a Server Side of a Public Network</i> Per Limitation 13.1	41
1.	Gupta Discloses a <i>Client-Side</i> User Interface Module 152 that Performs <i>Client-Side</i> Cue Point Editing	41
a.	<i>Client-Side</i> UIM 152 Application, Not Annotation Server 10, Creates and Edits Annotations in Gupta	45
b.	<i>Client-Side</i> UIM 152 Generates the Client-Side Dialog Boxes Including the “Edit Annotation” Figure 21 Used to Edit an Annotation	53
2.	Petitioner Aylo’s Argument for Limitation 13.1.....	62
3.	Gupta Does Not Disclose a Server-Side Application on Annotation Server 10 For Providing a <i>Video Editing Environment</i> per Limitation 13.1.....	64
4.	The Petitioner’s Expert’s Assertion that Annotation Server 10 Executes Software for Editing Annotations Is Contrary To the Evidence	66
5.	Gupta’s Use of HTTP Requests Sent from Client 15 to Annotation Server 10 Is Standard Client-Server Communications Protocol and Does Not Mean that Annotation Server 10 is Executing a <i>Video Editing Environment</i> Application.....	69

6.	Summary on Gupta's Failure to Anticipate Limitation 13.1.....	74
E.	Gupta Fails to Disclose <i>Modifying a Portion of the Cue Point Information at the Server Side ... According to the Cue Point Editing Command Received ... From the Client Terminal</i> (Limitation 13.5)	75
F.	Gupta Fails to Disclose Limitation 13.2	80
G.	Gupta Fails to Disclose Claim 15 For <i>Receiving an Edit Mode Entry Signal ... From the Client Terminal Representing Entry into the Cue Point Edit Mode, and Switching the Video Editing Environment to the Cue Point Editing Mode in Response to Receiving the Edit Mode Entry Signal</i>	81
H.	Gupta Fails to Disclose Claim 18 For <i>Receiving, Over the Network, Authentication Information Related to Access Authority of the Remote Client Terminal</i>	88
I.	Claims 14 and 17-20 Are Patentable Over Gupta.....	90
IX.	CLAIMS 13-15 AND 18-20 ARE NOT ANTICIPATED BY FEGAN (GROUND 2).....	90
A.	Claims 13-15 and 18-20 Are Patentable Over Fegan.....	90
B.	Fegan Fails to Disclose <i>A Computer System ... Comprising a Video Editing Environment, and Being Disposed at a Server Side of a Public Network Per Limitation 13.1</i>	91
1.	Fegan Does Not Disclose a <i>Server-Side Video Editing Environment Application Per Limitation 13.1</i>	91
2.	Fegan's Reference to MYSQL Does Not Disclose a <i>Server-Side Video Editing Environment Application Per Limitation 13.1</i>	93
3.	Fegan's References to Programs Stored on Remote Servers for the Search Window of Figure 2 Does Not Disclose a <i>Server-Side Video Editing Environment Application Per Limitation 13.1</i>	95

4.	Fegan’s Reference to Programs Stored on Remote Servers in Connection with the Search Results Window of Figure 3 Does Not Disclose a Server-Side <i>Video Editing Environment</i> Application Per Limitation 13.1	98
5.	Fegan’s Reference to “Software Stored on a Remote Server” to Save Segments and Annotations Does Not Disclose a Server-Side <i>Video Editing Environment</i> Application Per Limitation 13.1	100
6.	Fegan’s Segment/Annotation Creation Window of Figure 4 Based on the Client-Side Multimedia Player and the Client-Side Browser Does Not Disclose a Server-Side <i>Video Editing Environment</i> Application Per Limitation 13.1	102
a.	A POSITA Would Understand That Fegan’s “Browser-Based Multimedia Player” is Client Software	105
b.	Multimedia Players Were Understood in the Art as Client Applications.....	105
c.	Figure 4 Reflects Applications for Creating Annotations, Not Editing Annotations, And the Applications Are Client-Side, Not Server-Side.....	106
7.	Fegan’s Discussion of the Portal Window of Figure 5 Does Not Disclose a Server-Side <i>Video Editing Environment</i> Application Per Limitation 13.1	109
8.	Fegan’s Discussion of the Annotation View Window of Figure 6 Does Not Disclose a Server-Side <i>Video Editing Environment</i> Application Per Limitation 13.1	112
9.	Fegan’s Discussion of the Segment/Annotation Edit Window of Figure 7 Does Not Disclose a Server-Side <i>Video Editing Environment</i> Application Per Limitation 13.1.....	114
10.	Summary of Fegan Regarding Limitation 13.1 for a Server-Side <i>Video Editing Environment</i> Application.....	115

C.	Fegan Fails to Disclose <i>Receiving a Cue Point Editing Command ... From a Remote Client Terminal And Modifying a Portion of the Cue Point Information ... According to the Cue Point Editing Command</i> (Limitations 13.4-13.5).....	115
1.	Limitation 13.4 for <i>Receiving a Cue Point Editing Command</i>.....	116
2.	Limitation 13.5 for <i>Modifying ... the Cue Point Information ... According to the Cue Point Editing Command ... From the Client Terminal</i>.....	118
D.	Fegan Fails to Disclose Claim 15 for <i>Receiving an Edit Mode Entry Signal ... From the Client Terminal Representing Entry into the Cue Point Edit Mode, and Switching the Video Editing Environment to the Cue Point Editing Mode in Response to Receiving the Edit Mode Entry Signal</i>	119
E.	Fegan Fails to Disclose Claim 18 for <i>Receiving, Over the Network, Authentication Information Related to Access Authority of the Remote Client Terminal</i>.....	122
F.	Claims 14-15 and 18-20 Are Patentable Over Fegan.....	123
X.	CLAIM 17 IS PATENTABLE OVER FEGAN AND SULL (GROUND 2* FROM -01101 IPR)	123
XI.	CLAIMS 13-15 AND 17-20 ARE PATENTABLE OVER KIM (GROUND 3).....	124
A.	Claims 13-15 and 17-20 Are Patentable Over Kim.....	124
B.	Kim Fails to Meet Limitation 13.1 for <i>Providing a Computer System ... Comprising a Video Editing Environment, and Being Disposed at a Server Side of a Public Network</i>	124
1.	Kim’s Bookmarks Are Created Locally Using the Multimedia Bookmark Client 238 Running on Client 230.....	127
2.	Kim’s Server-Side Components Do Not Provide a Server-Side <i>Video Editing Environment</i> per Limitation 13.1	131

3.	Petitioner Aylo’s Arguments Fail to Establish Kim Meets Limitation 13.1 for a Server-Side <i>Video Editing Environment</i>	133
C.	Kim Fails to Meet <i>Receiving a Cue Point Editing Command ... From a Remote Client Terminal And Modifying a Portion of the Cue Point Information ... According to the Cue Point Editing Command (Limitations 13.4-13.5)</i>.....	137
1.	Limitation 13.4 for <i>Receiving a Cue Point Editing Command Over the Network from a Remote Client Terminal</i>	137
2.	Limitation 13.5 for <i>Modifying a Portion of the Cue Point Information at the Server Side ... According to the Cue Point Editing Command ... From the Client Terminal</i>	142
D.	Kim Fails to Meet Claim 15 for <i>Receiving an Edit Mode Entry Signal ... From the Client Terminal Representing Entry into the Cue Point Edit Mode, and Switching the Video Editing Environment to the Cue Point Editing Mode in Response to Receiving the Edit Mode Entry Signal</i>	142
E.	Kim Fails to Meet Claim 17 for <i>Generating, Automatically, By Using the Computer System ... A Portion of the Set of Cue Point Information</i>	146
F.	Kim Fails to Meet Claim 18 for <i>Receiving, Over the Network, Authentication Information Related to Access Authority of the Remote Client Terminal</i>	146
G.	Claims 14-15 and 17-20 Are Patentable Over Kim.....	147
XII.	CLAIMS 13-15 AND 17-20 ARE PATENTABLE OVER KIM AND GUPTA (GROUND 4).....	147
A.	Independent Claim 13 Is Patentable Over Kim and Gupta.....	147
B.	Kim and Gupta Fail to Teach or Suggest Claim 15 and Claim 18.....	150
C.	Claims 14-15 and 17-20 Are Patentable Over Kim and Gupta ..	151

XIII. CLAIMS 13-15 AND 17-20 ARE PATENTABLE OVER VASILEVSKY AND LOGAN (GROUND 5)	151
A. Claims 13-15 and 17-20 Are Patentable Over Vasilevsky and Logan	151
B. Vasilevsky and Logan Fail to Meet Limitation 13.1 for <i>Providing a Computer System ... Comprising a Video Editing Environment, and Being Disposed at a Server Side of a Public Network</i>	152
1. The Assertion that Vasilevsky’s Home Area Network Is a Public Network Is Not Credible	152
2. Vasilevsky and Logan Do Not Meet the Claimed <i>Video Editing Environment</i> Application Because Vasilevsky Does Not Disclose Any Software Application for Editing Bookmarks	153
3. The Petitioner’s Citations to Vasilevsky Fail to Teach the Claimed Server-Side <i>Video Editing Environment</i> Application.....	161
4. A POSITA Would Not Modify Vasilevsky based on Logan to Rehost Vasilevsky onto the Public Internet	166
a. Modifying Vasilevsky to Be Hosted on the Public Internet in Place of the Home Area Network Would Not Work	167
b. The Proposed Combination to Rehost the Alleged <i>Computer ... Comprising a Video Editing Environment</i> on the Internet Would Not Work Because There is No Front-End Receiver Circuitry to Enable Reception and Storage of Selected Programs as Required by Vasilevsky	173
c. There is No Apparent Reason to Combine As Proposed and There Would Not Be a Reasonable Likelihood of Success Because of the Deficiencies in the Combination	177

C.	Vasilevsky and Logan Fail to Meet Limitation 13.2 for <i>Receiving a Set of Video Information At the Video Editing Environment</i>	180
D.	Vasilevsky and Logan Fail to Meet Limitation 13.3 for <i>Receiving a Set of Cue Point Information At the Video Editing Environment</i>	183
E.	Vasilevsky and Logan Fail to Meet Limitations 13.4 and 13.5 for <i>Receiving a Cue Point Editing Command ... From a Remote Client and Modifying a Portion of the Cue Point Information at the Server Side ... According to the Cue Point Editing Command</i>.....	188
F.	Vasilevsky and Logan Fail to Meet Claim 15 for <i>Receiving an Edit Mode Entry Signal Over the Network From the Client Terminal Representing Entry into the Cue Point Edit Mode, and Switching the Video Editing Environment to the Cue Point Editing Mode in Response to Receiving the Edit Mode Entry Signal</i>	191
G.	Vasilevsky and Logan Fail to Meet Claim 17 for <i>Generating, Automatically, By Using the Computer System ... A Portion of the Set of Cue Point Information</i>	196
H.	Vasilevsky and Logan Fail to Meet Claim 18 for <i>Receiving, Over the Network, Authentication Information Related to Access Authority of the Remote Client Terminal</i>.....	197
I.	Claims 14-15 and 17-20 Are Patentable Over Vasilevsky and Logan	198
XIV.	CONCLUSION.....	198

I. INTRODUCTION

1. My name is Dr. Mark Mahon. I have been retained as an expert witness to provide my independent opinion in regards with matters at issue in the *inter partes* review of U.S. Pat. No. 8,307,286 (“the ’286 Patent”) in IPR2024-00710 proceeding. I have been retained by WellcomeMat, LLC (“WellcomeMat” or “Patent Owner”), the Patent Owner, in this proceeding. The Petitioner in this case is Aylo Freesites Ltd (“Aylo” or “Petitioner”).

2. Unless otherwise noted, the statements made herein are based on my personal knowledge, and if called to testify about this declaration, I could and would do so competently and truthfully. Unless otherwise noted or apparent from the context, it should be understood that my discussion of the ’286 Patent and the prior art is presented from the viewpoint of a Person of Ordinary Skill in the Art (“POSITA”) as of the priority date of the ’286 Patent.

3. I am being compensated for my work in this matter at the rate of \$550/hour. I am also being reimbursed for reasonable and customary expenses associated with my work and testimony in this investigation. My compensation is not contingent on the outcome of this matter or the specifics of my testimony.

4. I am not a legal expert and offer no opinions on the law. However, I have been informed by counsel of the various legal standards that apply, and I have applied those standards in arriving at my conclusions.

5. In this declaration, I have been asked to address certain invalidity opinions relating to the '286 Patent, specifically those provided by Aylo in its first Petition, challenging Claims 13-15 and 18-20 (IPR2024-00710) and its second Petition (IPR2024-01101) challenging Claim 17 (collectively, the "Petition") and the accompanying declaration of Dr. Henry Houh, Ex. 1002 ("Houh Decl." or "Houh declaration"), regarding the alleged invalidity of the '286 Patent.¹ The Petition and Dr. Houh argue that Claims 13-15 and 17-20 ("Challenged Claims") are unpatentable.

6. In this expert declaration, I provide my opinions in response to the Grounds in the Petition and the Houh declaration, as well as the basis for my opinions. As I explain in detail later in this expert report, based on my study of the '286 Patent, the Houh declaration, and various references cited by Petitioner and Dr. Houh, it is my opinion that the Challenged Claims are patentable over the cited references alone or in the various combinations identified in the Grounds.

7. I have found multiple limitations in the Challenged Claims that are not disclosed by any one of those references or a proposed combination of references. I identify those undisclosed limitations on a claim-by-claim, reference-by-reference

¹ The "Houh declaration" is understood to refer to the declaration submitted in IPR2024-00710 unless otherwise noted.

basis. I also identify proposed combinations of references in a Ground or proposed modifications of a reference in a Ground that fail to render claims obvious because a person of ordinary skill in the art (“POSITA”) would not have an apparent reason to combine or modify the references as proposed by the Petitioner and Dr. Houh. That said, my decision not to address a limitation of a Challenged Claim should not be interpreted as an admission that such a limitation is found in the art in the Ground cited by Petitioner and Dr. Houh, or that a POSITA would have an apparent reason to combine or modify references in a manner meeting the limitation.

8. I submit this declaration having reviewed the ’286 Patent, the Challenged Claims, the ’286 Patent file history, and various references cited by Petitioner in its Petition and/or in the Houh declaration. The material I have considered includes but is not limited to the items set forth below:

PETITIONER MATERIALS CONSIDERED	
Paper 1	Petition in IPR2024-00710, -01101
Paper 13	Institution Decision dtd 9/5/2024
Ex. 1002	Declaration of Dr. Henry Houh
Ex. 1003	U.S. Patent Application Publication No. 2005/0081159 (“Gupta”)
Ex. 1005	U.S. Patent Application Publication No. 2005/0144305 (“Fegan”)

Ex. 1006	U.S. Patent Application Publication No. 2005/0210145 (“Kim”)
Ex. 1007	U.S. Patent Application No. 09/911,293; Publication No. 2002/0069218 (“Sull”)
Ex. 1008	WO 2003/067594 A1 (“Vasilevsky”)
Ex. 1009	U.S. Patent Application Publication No. 2003/0093790 (“Logan”)
Ex. 1010	Prosecution History for U.S. Pat. No. 8,307,286
Ex. 1017	Excerpts from Microsoft Computer Dictionary, 5th Ed. (2002)
Ex. 1018	“Signal.” Merriam-Webster.com Dictionary, Merriam-Webster, https://www.merriam-webster.com/dictionary/signal (accessed March 29, 2024)

9. I have also considered materials from the following list of Patent Owner’s list of exhibits:

PATENT OWNER MATERIAL CONSIDERED	
Ex. 2012	What is MYSQL (accessed at oracle.com on 6/23/2024)
Ex. 2016	WO2005/071962 to Wasilewski (Scientific-Atlanta)
Ex. 2017	US20030140353A1 to Hugenberg (Quest Communications)
Ex. 2019	U.S. Pat. No. 10,631,026 to Carlucci (Time Warner Cable)
Ex. 2020	U.S. Pat. No. 4,694,490 to Harvey

Ex. 2021	U.S. Pat. No. 7,269,841 to Hendricks
Ex. 2022	Vasilevsky and Logan
Ex. 2023	HTTP – Wikipedia
Ex. 2025	Computer Networks by Tanenbaum (2003)
Ex. 2026	U.S. Pat. No. 6,094,677 (Capek) to IBM
Ex. 2027	U.S. Pat. No. 7,216,300 (Dang) to NCR
Ex. 2028	Java Applet – Wikipedia
Ex. 2029	U.S. Pub. No. 2009-0046147A1 (Henson) to Safehouse Int'l
Ex. 2030	U.S. Pat. No. 6,691,070 (Williams) to Mack Info. Systems
Ex. 2031	U.S. Pub. No. 2003-0182139A1 (Harris) to Microsoft
Ex. 2032	WO2005122025A2 (Krikorian) to Sling Media
Ex. 2033	U.S. Pat. No. 7,797,242 (Gautier) to Apple
Ex. 2034	Decision on Institution, Paper 13, 9/5/2024
Ex. 2036	U.S. Pub. No. 2006/0150031 (Seiko)
Ex. 2037	U.S. Pub. No. 2003/0131052 (IBM)
Ex. 2038	U.S. Pat. No. 6,874,021 (Cisco)
Ex. 2039	U.S. Pat. No. 7,283,532 (Hassan-Ali to Alcatel)

Ex. 2040	U.S. Pat. No. 6,950,968 (Parolari)
----------	------------------------------------

10. I have also considered other materials, such as web sites and text materials in doing background research on the issues presented in this IPR.

11. My opinions as set forth in this expert declaration are based on these sources of information, together with my education, training, and experience, including my work in the field of the '286 Patent.

II. QUALIFICATIONS AND PROFESSIONAL EXPERIENCE

12. My complete qualifications and professional experience are described in my Curriculum Vitae, a copy of which is submitted as Exhibit 2014 to the Patent Owner's Response in this IPR.

13. The following is a brief summary of my relevant qualifications and professional experience.

14. I am a Teaching Professor in the School of Electrical Engineering and Computer Science at Pennsylvania State University, University Park, PA ("Penn State" or "PSU"). In the early to mid-1990's, a portion of my research work and field testing focused on acoustic detection, localization, automated tracking, event detection, and automated notification systems for commercial and Department of Defense (DoD) entities integrated with video and still imagery. Additionally, I have worked on telecommunications and wireless networks, including Z-Wave,

Bluetooth, Zigbee, Wi-Fi, NFC, AMPS, IS-95, CDMA2000, GSM, EDGE, UMTS/WCDMA, LTE, and 5G cellular systems since 1988.

15. I received my B.S. in Electronics Engineering from the University of Scranton in 1987. I received my M.S. in Electrical Engineering and Ph.D. in Acoustics from Penn State in 1991 and 2001, respectively.

16. In 1988, after I received my bachelor's degree, I joined the Central Intelligence Agency (CIA) while pursuing my M.S. degree at Penn State part-time. My first job at the CIA involved designing and testing systems to automatically capture and characterize telecommunication signals and emissions from various wireless and computer networking devices.

17. I returned to Penn State in early 1990 to pursue graduate research full-time and complete my M.S. degree. My graduate research work focused on wideband beamforming and adaptive signal processing. After completing my M.S. degree in EE in 1991, I accepted a full-time faculty research position at the Applied Research Lab at PSU, primarily working on classified programs, and began working on diverse radio frequency and acoustic sensor systems including wireless communications and small wireless networks for acoustic tracking, source localization, and feature extraction.

18. I began pursuing my Ph.D. in 1993 while continuing my faculty research position. The research work and field testing I carried out between 1991

and 1999 involved detection systems for queuing autonomous tracking and notification. This included video and still imagery components integrated with detection events to automatically record acoustic and visual events, apply metatags associated with time and location, and sending notifications which included detection associated information for reachback by a responsible person to verify and/or investigate the event. These notifications used wired and wireless networks as well as telephone lines (traditional landlines). In 1997, as part of my faculty research position, I began working on classified programs focused on mathematical analytical modeling of cellular communication networks and the development of hardware and software systems to test against cellular networks. My role was to develop the algorithms and write the code running on a specially developed embedded system. For this work, I received a letter of recognition as the “genius behind the VELA software algorithms” from the Director of National Reconnaissance Office (NRO) Systems Engineering and Technology Office. As part of this same work, I was extensively involved in protocol and signaling analysis as well as researching model-specific performance and unique functional characteristics associated with individual mobile devices. The work involved testing dozens of handsets from many manufacturers in controlled and real-world environments against network simulators and live operational networks for each research project.

19. In 2000 my research extended into utilizing non-orthogonal wavelets for improving detection and localization of cellular handsets from high altitude sensor systems. In 2001, I completed my Ph.D. and my research focused on the utilization of advanced communication signals for wideband characterization and remote sensing of propagation channels.

20. Beginning in 1997 my cellular communications research work focused on CDMA, GSM, EDGE, UMTS, LTE, and 5G cellular systems primarily under grants sponsored principally by the Department of Defense. This classified research work required 3GPP protocol analysis and development of real-time embedded hardware and software systems capable of interacting with cellular networks and cellular handsets. A large portion of my work was directed at architectures, protocols, software, and signaling.

21. I have been working on classified projects since 1988. Before 1998, because the work was not deemed highly classified, I was able to publish eight journal and conference papers prior to 2000. Between 1999 and 2015, however, I was allowed to publish only one article in an unclassified symposium. I published and presented about a dozen articles in classified settings. This is because during this period, the vast majority of my research was highly classified. As a result, nearly all of my research results were summarized in classified reports and not available to the general public. Further, because the U.S. government owns any intellectual property

resulting from the sponsored research work, I did not pursue or file patent applications.

22. Beginning in 2003, I was co-principal investigator and technical lead on a 3-year multi-million-dollar research effort for developing the Global Information Grid (GIG). This project was sponsored by the Secretary of Defense's Office with a goal of developing a real-time, multi-intelligence (multi-Int) network for collecting, processing, storing, disseminating, and managing information on demand for decision makers including the warfighter, combatant command centers, policy makers, and support personnel and was the largest network-centric warfare project in development at the time. My research team (Ubiquitous Automated Information Manager) focused on building and deploying a scalable application to perform real-time, multi-int data fusion to support every user in the system. The fused data included live video feeds, recorded videos, and still imagery. Our application interacted with the video and imagery source feeds, consumed associated metadata, and implemented fusion algorithms which would, for example, direct intelligence analysts to specific time/event-tagged positions in the videos. Analysts could also utilize our software to augment or insert notes and/or tags for consumption by other analysts utilizing the system. This software application was deployed in Combat Operation Centers, Joint Interagency Task Force locations, and on various platforms (mobile and small computing environments) used by various

warfighters. The fused data sources included various content management systems, supply chain logistic reports, GPS-based reports, new feeds, backend databases, sensor system reports, and various other broad data sources.

23. In 2015, I transferred to the School of Electrical Engineering and Computer Science at Penn State as a teaching faculty member. In that role, I have continued teaching graduate and undergraduate courses, guiding Ph.D. and M.S. students in communication and mobile networking (including LTE and 5G cellular networks), and pursuing research in this and related areas. Since 2015, I have been an author on nine refereed papers as listed in my CV.

24. Because of my decades of research and my continuing work at Penn State, I have intimate knowledge of telecommunication networks. The U.S. government awarded me over \$12M in grants between 2003 and 2015 for projects focused on mobile communication devices and networks, in which I served as a Principal Investigator (PI), Co- PI, and/or technical lead.

25. Additionally, during my research career, I interacted extensively with computer scientists and engineers responsible for the design, development, and testing of telephony and data networking systems and testbeds. As a research faculty member, I oversaw engineers and computer scientists that executed many joint projects with development organizations. These interactions exposed me to a wide range of computer scientists and engineers working on telecommunication network

technologies. Since 2011, I have been teaching undergraduate and graduate classes in communication and mobile networking and am familiar with the curricula being taught to electrical engineers and computer scientists.

26. While most of my research efforts between 1998 and 2015 were highly classified, I can state that they included detailed investigation of network architectures, signaling, and functional behavior. A typical research effort would involve studying 3GPP, 3GPP2, IEEE, and other protocol standards to fully comprehend all aspects of L1, L2, and L3 requirements including timing, bit-level construction of the control and user plane messages, and timing characteristics for a given standard as well as functional behavior of network components and user equipment.

27. From 2006 through 2015 my research focused specifically on LTE. My research continues to this day, although I am no longer operating in a classified environment. During this time, I investigated the performance and functional differences of many varied network and handset devices to see how differing signaling and hardware configurations (including MIMO) and environmental factors influenced the behavior of user equipment in a given network environment. This included how diversity techniques (transmit and receive), synchronization, timing, and signal to interference plus noise ratio (SINR) for a given device would affect specific functional aspects including elements of the receiver structure, decoding and

demodulation performance, calculation of parameters used by the device for making decisions and deriving parameters reported to the network.

28. As part of my research work, I built several custom CDMA, GSM, UMTS, and LTE platforms that implemented specific network-side and user equipment-side functionality including custom signal generation and processing structures, particularly the signal processing chains on both the transmit and receive sides. This equipment was developed using network simulation hardware in a laboratory environment and was later tested with corresponding networks in both controlled and fully operational environments. Implementing the transmit and receive chains for custom built protocol-enabled equipment required me to gain an intimate understanding of the relevant 3GPP protocol specifications and the underlying structures. Since 2015, I have been primarily focused on guiding graduate students pursuing research including using code domain non-orthogonal multiple access (NOMA) combined with MIMO sparse coding multiple access to minimize latency and maximize user density in grant free Internet of Things (IoT) environments. Additionally, I am guiding my graduate students in pursuing research in optimized distributed processing algorithms, implementation of block chain coding techniques to improve handover security, and edge computing resource allocation in 4G (LTE)/5G (NR) networks.

29. Much of the classified research work I performed also led to similar approaches for other wireless protocols including IEEE 802.11 and 802.15 (e.g., Zigbee, Bluetooth, and UWB), HART, and other short-range standards as well as HF radio and Wi-MAX.

30. Over the course of my career, I have accumulated significant expertise in processing video and still image data, including creating annotations of video and the distribution of video over networked systems, as well as navigating video.

III. RELEVANT LEGAL STANDARDS

A. Burden of Proof

31. I understand that Petitioner Aylo has the burden of proving unpatentability by a preponderance of the evidence. 35 U.S.C. § 316(e). Preponderance of the evidence “means the greater weight of evidence, evidence which is more convincing than the evidence which is offered in opposition to it.” *Hale v. Dep’t of Transp.*, 772 F.2d 882, 885 (Fed. Cir. 1985).

B. Anticipation (35 U.S.C. § 102)

32. I also understand that a patent may be rendered invalid or unpatentable if it is anticipated based on an alleged prior art reference as understood by a person of ordinary skill in the art (POSITA). Anticipation requires that a prior art reference disclose the claimed invention in its totality, that is, all limitations of the claim are disclosed in the reference as set forth by the claim.

33. I understand that anticipation is an exacting standard. “A claim is anticipated if each and every limitation is found either expressly or inherently in a single prior art reference.” *King Pharm., Inc. v. Eon Labs, Inc.*, 616 F.3d 1267, 1274 (Fed. Cir. 2010) (quotations omitted). “But disclosure of each element is not quite enough—this court has long held that ‘[a]nticipation requires the presence in a single prior art disclosure of all elements of a claimed invention *arranged as in the claim.*’” *Net MoneyIn v. Verisign*, 545 F.3d 1359, 1371 (Fed. Cir. 2008) (cited omitted) (orig. italics). I understand that to mean that the reference must disclose the invention exactly as it is claimed.

C. Obviousness (35 U.S.C. § 103)

34. I also understand that a patent may be rendered “obvious” based on an alleged prior art reference or a combination of such references in view of what a person of ordinary skill in the art would understand based on his or her knowledge and the references. I understand that a patent cannot be properly granted for subject matter that would have been obvious to a person of ordinary skill in the art at the time of the alleged invention, and that a patent claim directed to such obvious subject matter is invalid under 35 U.S.C. § 103. It is also my understanding that in assessing the obviousness of claimed subject matter one should evaluate obviousness over the prior art from the perspective of one of ordinary skill in the art as of the priority date

of the invention (and not from the perspective of either a layman or a genius in that art).

35. I have been informed that the following factors must be evaluated to determine whether the Petitioner has met its burden of proof that a claimed invention is obvious:

1. The scope and content of the prior art relied upon by Petitioner;
2. The difference or differences, if any, between each claim of the patent and the prior art;
3. The level of ordinary skill in the art at the time the invention of the patent was made (e.g., the priority date); and
4. Any objective indicia of non-obviousness, such as: (1) commercial success of an embodiment; (2) a long-felt need; (3) skepticism; (4) failure by others to find the solution provided by the claimed invention; (5) copying by others of the subject matter of the claim invention; (6) unexpected results of the claimed invention; (7) acceptance of others and industry praise; and (8) licensing of the patents.

36. I have been informed that to prevail on obviousness the Petitioner must establish that (1) the prior art teaches or suggests each of the limitations of the claim; (2) there exists an apparent reason to combine and/or modify the prior art as proposed; and (3) a person of ordinary skill would have a reasonable expectation of

success that the proposed combination and/or modification of the prior art would operate for its intended purpose.

37. I have been informed that a claim is not proved obvious merely by demonstrating that each of the elements was independently known in the prior art. I have been informed that many, if not all, inventions rely on building blocks already known, and that claimed inventions almost of necessity will likely be combinations of what is already known.

38. I have been informed that it is important in the obviousness inquiry to identify whether a POSITA at the time of the invention would have motivation to combine or modify the prior art references in the manner proposed by the Petitioner so as to arrive at the claimed invention. Putting it another way, a finding of obviousness should be supported by a showing that a POSITA would have an apparent reason to combine or modify the prior art references as proposed by the Petitioner.

39. I have been informed that it is important in the obviousness inquiry that the Petitioner explains how the references are combined, including an explanation of how the combination of references is supposed to work. An explanation of the operation of the combined references is often a prerequisite to determining whether a POSITA would have been motivated or had a reason to make the proposed combination/modification and would have had a reasonable expectation of success.

40. In assessing obviousness, I have been instructed to consider both the ordinary creativity and common sense of the person of ordinary skill in the art. I also understand that it is normally impermissible for common sense or common knowledge to be applied to fill gaps in prior art that fails to teach or suggest a limitation of the claim. I also understand that the obviousness inquiry should guard against hindsight bias or hindsight reconstruction where after-the-fact reasoning is applied to combine prior art elements using the claim as a template, without establishing that, as of the date of the invention (e.g., the priority date), there exists a motivation to combine or apparent reason to combine or modify the prior art as proposed.

41. In assessing obviousness, I have been instructed that, in order to qualify as proper prior art for an obviousness analysis, a reference must qualify as analogous art. I have been informed that a reference qualifies as analogous art with respect to the claims if it is either: (1) from the same field of endeavor as the patent; or (2) the reference is reasonably pertinent to the particular problem addressed by the invention. I have also been informed that for a reference to be reasonably pertinent, it must logically have commended itself during the ordinary course of development to an inventor's attention in considering the problem.

42. In assessing obviousness, I have also been instructed that, as noted above, secondary considerations can often be probative evidence of non-obviousness

and can serve to avoid hindsight bias. I have been instructed that secondary considerations can be sufficient to overcome a prima facie case of obviousness.

IV. LEVEL OF ORDINARY SKILL IN THE ART AND PRIORITY DATE

43. My opinions in this declaration are based on the understandings of a person of ordinary skill in the art, which I understand is sometimes referred to as an “ordinary artisan” or by the acronym “POSITA” (Person of Ordinary Skill in the Art), as of the time of the invention (e.g., the priority date). I understand that the person of ordinary skill in the art is a hypothetical person who is presumed to have known the relevant art at the time of the invention. By “relevant,” I mean relevant to the Challenged Claims of the ’286 Patent.

44. For purposes of this Declaration, I have been instructed to use May 7, 2006, as the time of invention (priority date). I understand that the Petitioner does not dispute the May 2006 priority date. See Ex. 1002 (Houh Decl.), ¶¶ 57-58.

45. I understand that, in assessing the level of skill of a person of ordinary skill in the art, one should consider the field of endeavor, the type of problems encountered in the art, the prior solutions to those problems found in the prior art, the rapidity with which innovations are made, the sophistication of the technology, the level of education of active workers in the field, and my own experience working with those of skill in the art at the time of the invention.

46. In this IPR, the Petitioner and its expert submit that a person of ordinary skill in the art would have had a bachelor's degree in electrical engineering, computer engineering, or computer science (or equivalent) plus approximately two years of work experience with technologies relating to video annotation/bookmarking and/or video navigation. Pet., 15; Houh Decl., ¶¶ 59-60.

47. I believe that definition of the level of skill of the POSITA is reasonable, except that I would add that lack of work experience could be mitigated by additional education, and vice versa.

V. THE '286 PATENT

A. Procedural History

48. The original Petition for the -00710 IPR challenges Claims 13-15 and 18-20 of the '286 Patent as obvious based on five grounds. The follow-on petition filed for IPR2024-01101 challenging Claim 17 sets forth the same grounds (combinations of references and anticipation or obviousness theory) except for one different combination of references. The -01101 IPR was terminated and consolidated with the -00710 IPR in an order issued on December 6, 2024. See Order (IPR2024-01101, Decl. 6, 2024), Paper 10, 2-3; Order (IPR2024-00710, Decl. 6, 2024), Paper 18, 2-3.

49. With the proceedings consolidated, the following grounds are presented:

- Ground 1 challenging Claims 13-15 and 17-20 as anticipated under § 102 based on Gupta;
- Ground 2 challenging Claims 13-15 and 18-20 as anticipated under § 102 based on Fegan;
- Ground 2* challenging Claim 17 as obvious under § 103 based on Fegan and Sull;
- Ground 3 challenging Claims 13-15 and 17-20 as obvious under § 103 based on Kim;
- Ground 4 challenging Claims 13-15 and 17-20 as obvious under § 103 based on Kim and Gupta; and
- Ground 5 challenging Claims 13-15 and 17-20 as obvious under § 103 based on Vasilevsky and Logan.

See Pet. (IPR2024-00710), 1; Pet. (IPR2024-01101), 1. Each of the above grounds is at issue in consolidated IPR2024-00710.

B. Overview of the '286 Patent

50. U.S. Pat. No. 8,307,286 (“the ’286 Patent”) is directed to a system and method for providing video content supplemented with cue point information. ’286 Patent, 2:51-3:3. The cue point information can be used to explain the video with comments and other supplemental information. The cue point information also provides a mechanism to quickly navigate the video. For example, a real estate

listing video presenting a home for sale can have cue points marking locations for video segments depicting the foyer, living room, neighborhood, and exterior of the home. '286 Patent, Fig. 6. Rather than view the entire video, the user can navigate to specific segments by selecting cue points that have been inserted.

51. Prior art systems for processing cue point information relied on specialized client applications that were cumbersome and difficult to learn. '286 Patent, 2:16-25. A key innovation of the '286 Patent is that it enables cue point management to take place remotely over a public client-server architecture, which increases the flexibility and manageability of cue points and allows for modifications to be made to content that has already been published and distributed.

52. For example, video might depict a home for sale that has been listed on multiple locations on the internet (e.g., MLS, Trulia, Zillow, Realtor.com, etc.). Suppose that the videos need to be updated to include a cue point for a segment showing the backyard. Using the invention of the '286 Patent, the user can perform remote cue point editing to edit the video at an intermediary video production website without the client having to download and locally edit each of the videos using a client-side application to insert the cue point. This is a dramatic improvement over prior art systems that at the time required cue point management to be performed locally using a local client application. Such prior art systems were not

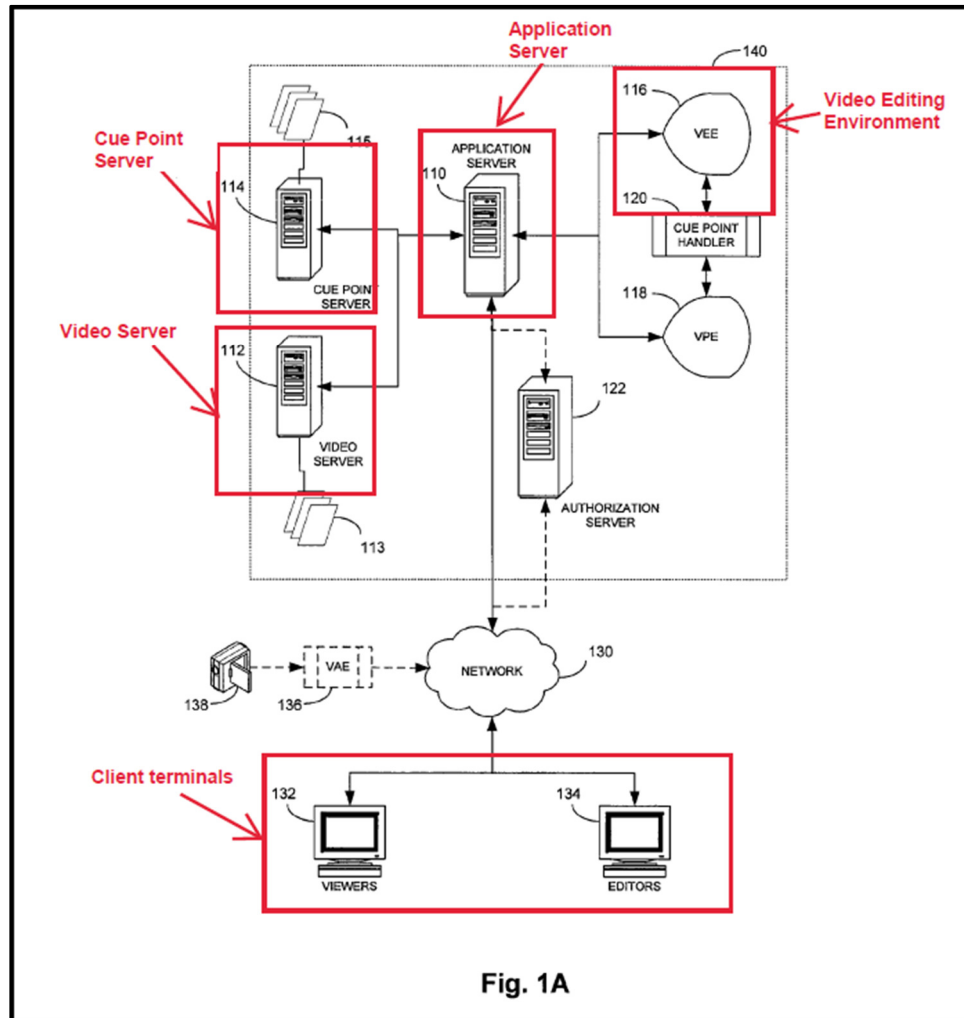
capable of modifying cue point information for video content that had already been released.

53. The '286 Patent describes and claims a specialized server-based architecture that enables “remote editing of and/or interaction with cue points associated with a video file over a public client-server architecture.” Ex. 1010 ('286 Patent File History), 112. In other words, the invention provides a server-side video editing environment providing for remote cue point editing by a client device. In contrast, prior art techniques and systems permitted only local cue point editing at a client device using client-side applications. *Id.*

54. The cue point information is typically indexed to a specific time or location in video data and may include the index information, as well as other information such as metadata describing the cue point (e.g., the title of the cue point, the author of the cue point, etc.) and the cue point content (e.g., text, audio, video, etc.) that, for example, describes or explains the associated video. '286 Patent, 2:14-32, 7:49-8:38, 8:25-34, Fig. 6 (cue point comments 636). The cue points can be selected by a user to navigate to a portion of the associated video. '286 Patent, 15:18-21.

55. As illustrated in Figure 1A below, one embodiment of the cue point processing system includes an application server 110 that interfaces with a video server 112 containing video data and a cue point server 112 that contains sets of cue

point information corresponding to the video. '286 Patent, Fig. 1A. The application server 110, video server 112, and cue point server 114 are all server-side systems that interface over a public network with client terminals such as viewer terminals 132 or editor terminals 134.



'286 Patent, Fig. 1A (annotated).

56. As reflected in Figure 1A above, the application server runs a specialized server-side application(s), including a Video Editing Environment (VEE) application 116 and a Video Playback Environment (VPE) application 118.

Id., 9:19-25 (“application server 110 may be configured to handle various applications, including a Video Editing Environment (VEE) 116 and/or a Video Playback Environment (VPE) 118. These programs may be stored on the application server 110 ... able to be executed ... by the application server 110”), Fig. 1A. See 2:59-60 (“video environment is configured to be runnable by the application server”), 4:57-59, 6:64-66 (“application server 110 configured to execute, and possibly store, one or more applications”).

57. The VEE corresponds to one or more server-side applications executed by the application server to permit client devices (e.g., editor terminals) to remotely edit sets of cue point information associated with video data. ’286 Patent, 4:1-2 (“the video editing environment comprises a server-level video editor”), 13:2-6 (“the VEE ... load[s] as a ... server-level application”); 13:64-65 (“the VEE itself running at the ... server level”). “[T]he cue point information form may be displayed on a client terminal while the cue point editor resides on a server terminal.” ’286, 23:18-20. “[T]he [server-side] VEE may provide interactive editing capability.” *Id.*, 15:25-26. The server-side VEE may include a “cue point edit mode” entered upon receipt of a signal from the client device. *Id.*, 3:48-61, 15:26-39, Fig. 3A (320), 18:18-41, Fig. 3B (354).

58. The ’286 Patent refers to the VEE as a server-side application in connection with describing mechanisms that determine whether a client device can

access the VEE. '286 Patent, 13:16-24 (“For example, it may be desirable to limit access to the VEE only to certain types of clients. Similarly it may be desirable to open access to the VEE, but limit access to certain files. ... In these embodiments, the method 300 may determine based on the login information whether the client has permission 304 to access all or part of the VEE”).

59. The user may interface with the server-side VEE through a GUI provided, for example, “as a webpage on a client browser,” “a frame within a webpage,” or a “background script to be executed by another application.” '286 Patent, 13:59-67 (controlling server-side VEE with client-side browser).

60. The VEE may support transitioning to a cue point edit mode. '286 Patent, 3:33-61, 15:33-41 (“If the cue point edit mode has been entered ...”), 18:45-46 (“the client may select a GUI control dedicated to providing entry into the cue point edit mode”), 19:7-8 (“The method 370 begins by loading a cue point editor 372”), Figs. 3A-3C. According to an embodiment, the cue point editor permits sets of cue points (multiple cue points) for video to be edited as a group. '286 Patent, 15:61-16:4, 16:42-45 (“the client may be able to edit multiple cue points at once.”), Figs. 3A-3B. After video information and a set of cue points are loaded, “cue point information can be entered for new cue points or for modifying existing cue points.” '286 Patent, 16:5-6, Figs. 3A-3B.

61. The cue point edit mode may be entered in response to a signal transmitted from a client device to the server-side VEE:

In some embodiments, the method further provides a cue point editing mode configured to perform at least one of creating, modifying, deleting, or interpreting at least a portion of the set of cue point information. In certain of these embodiments, the method further **receives a pause signal over the network** representing that playback of the client video data has been paused, and **transmits [into] the cue point editing mode after receiving the pause signal**. In certain others of these embodiments, the method further **receives a pause signal** over the network representing that playback of the client video data has been paused, receives a time position representing the time position within the client video data at which playback was paused, and creates a cue point location associated at least in part with the client video data and the time position.

'286 Patent, 3:48-61 (emph. added) See also '286 Patent, 15:25-39 (polling for a signal input from a mouse or keyboard in Fig. 3A, 320), 18:18-46 (Fig. 3B, 354).

62. As particularly regards Claim 17, the '286 Patent also discloses embodiments whereby cue point information can be automatically generated. '286 Patent, 8:1-13 (“automatically create a cue point” based on audio recognition or video recognition), 21:59-61 (“one or more cue points may automatically be identified (e.g. by audio or video recognition)”); 22:27-40 (“receive cue point information” from “computer (e.g. automated or artificial intelligence)”).

63. Figure 6 depicts an exemplary client user interface for accessing the server-side Video Editing Environment (VEE) program. This user interface allows the client-side user to view video, display existing cue points for that video, navigate the video using the cue points, insert new cue points in the video, and modify existing cue points for the video. '286 Patent, 25:64-28:60, Fig. 6.

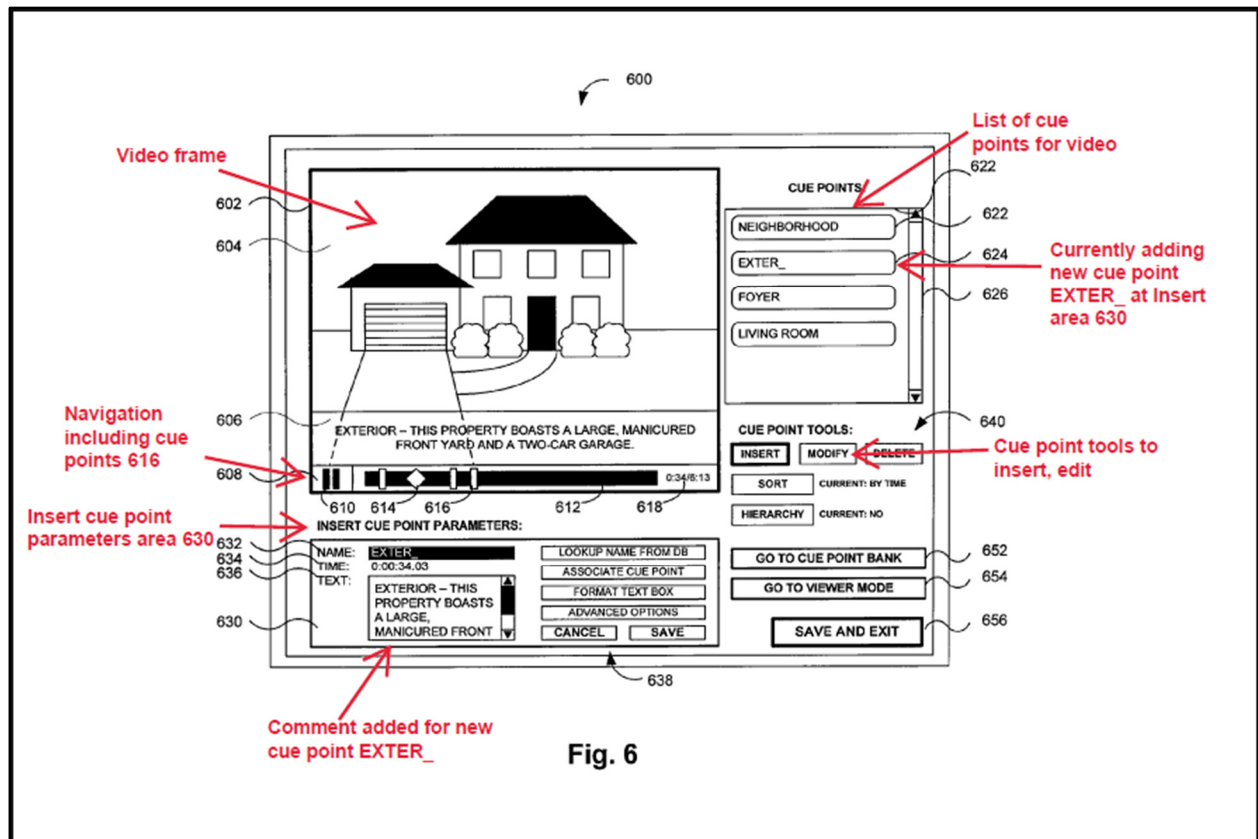


Fig. 6

'286 Patent, Fig. 6 (annotated in red).

C. Prosecution History

64. The application leading to the '286 Patent was filed on May 4, 2007, and asserts priority to a provisional application filed on May 7, 2006. Ex. 1001 ('286 Patent) (Cover, 60).

65. The application for the '286 Patent was subjected to a lengthy examination, receiving three office actions before the claims were allowed.

66. During the course of examination, the inventor repeatedly emphasized that the claimed invention distinguished the prior art by providing a decentralized architecture with a Video Editing Environment as a **server-side program** that allows the client terminal to remotely edit cue points over a public network, thus providing for **remote cue point editing** by the client, as opposed to local cue point editing using a local client application. See Ex. 1010 ('286 Patent File History), 111-113. For example, the inventor explained (all emph. added):

- “[T]he seller can ... create and modify those cue points **remotely**, via a remote video [editing] environment **running on an application server ...**” *Id.*, 112.
- “Accordingly, even if, arguendo, Shibata could be construed as teaching the cue point handling functionality recited in the claims, it would apparently only teach **local cue point handling ...** at the client terminal performing the editing. This is different from **remote editing** of and/or interaction with cue points associated with a video file **over a public client-server architecture.**” *Id.*, 112.
- “None of the cited art, alone or in combination, apparently teaches or suggests **remote editing** of and interaction with cue points associated

with a video file **over a public client-server architecture.**” Ex. 1010 (’286 Patent File History), 111.

- “In particular, both [prior art] implementations clearly describe **local editing functionality**, and only appear to use the network for distribution of content. For example, even though the plug-in application communicates with a host, it is still installed locally as a client plug-in to facilitate **local editing** and playback functions.... The combination would still seem to fail to teach or suggest the type of **remote cue point handling** recited in the claims.” Ex. 1010 (’286 Patent File History), 113.

67. The ’286 Patent issued on November 6, 2012.

D. The Claimed Invention

68. According to independent Claim 13, the invention provides a method for handling video cue points, comprising: [13.1] providing a computer system having at least one computer readable medium comprising a video editing environment, and being disposed at a server side of a public network; [13.2] receiving a set of video information at the video editing environment using the computer system; [13.3] receiving a set of cue point information at the video editing environment using the computer system; [13.4] receiving a cue point editing command over the network from a remote client terminal disposed at a client side of

the public network; [13.5] modifying a portion of the cue point information at the server side of the public network according to the cue point editing command; [13.6] receiving a cue point selection over the network from the remote client terminal identifying one of the cue points corresponding to a portion of the video information of one of the video files; and [13.7] streaming the portion of the video information to the client terminal over the public network in response to receiving the cue point selection.

VI. CLAIM CONSTRUCTION

69. I understand that claims of the patent-at-issue in this IPR are generally interpreted according to their ordinary and customary meaning in the art taking into consideration the so-called “intrinsic evidence” of the patent consisting of (1) the claim language; (2) the specification; and (3) the prosecution history.

70. I understand that claim terms may be explicitly defined in the patent specification, or they may be implicitly defined through consistent usage in the specification. I also understand that the scope of claim terms may be limited by statements in the specification or prosecution history where the applicant clearly disavows or disclaims subject matter, acts as his/her own lexicographer, or otherwise indicates the meaning of claim terms through statements made by the applicant.

71. I understand that for purposes of the institution stage, the Board found there was no need to provide express constructions of any claim terms. Instn. Decl., Paper 10, 10.

72. I have applied the ordinary and customary meaning of the claim terms in the context of the '286 Patent in establishing the patentability of the claims. This includes limitation 1.1 for *a computer system ... comprising a video editing environment, and being disposed at a server side of the public network*. As I explain in Section VIII.C, the ordinary and customary meaning of the claim language is informed by the specification and prosecution history.

73. I have also applied the ordinary and customary meaning of limitation 13.5 for “*modifying a portion of the cue point information at the server side of the public network according to the cue point editing command received ... from the client terminal.*” The Petitioner and its expert take the position that this limitation does not encompass storing the cue point information that was edited pursuant to the cue point editing command. Pet., 25 n.3; Ex. 1002 (Houh Decl.), ¶ 104. I disagree with this construction and address it in my analysis of the Ground 1 (Gupta) below.

VII. SUMMARY OF THE PRINCIPAL APPLIED REFERENCES

A. Gupta (Ex. 1004)

74. Gupta, entitled “User Interface for Creating Viewing and Temporarily Positioning Annotations for Media Content,” is generally directed to a client-server

system in which segments of multimedia content are associated with “annotations” (bookmarks) that are created using a User Interface Module (UIM) 152 running on a client device 15. The client-side-created annotations are transmitted to an annotation server 10, which receives them and stores the annotation content in a data store 17 and stores the annotation metadata in a data store 18. Gupta, Abstract, Figs. 1, 3.

B. Fegan (Ex. 1005)

75. Fegan, entitled “Systems and Methods for Identifying, Segmenting, Collecting, Annotating, And Publishing Multimedia Materials,” is generally directed to systems and methods for creating segments of video content and creating annotations (bookmarks) associated with the segments. Fegan, Abstract, [0003].

C. Kim (Ex. 1006)

76. Kim, entitled “Delivering and Processing Multimedia Bookmark,” is generally directed to a system allowing clients to locally create multimedia bookmarks associated with corresponding multimedia content files. The locally-created bookmarks are locally-stored at the user’s client device and, if the user elects, can be uploaded to a BBS server so that users can share bookmarks. Ex. 1006 (Kim), Abstract, Figs. 1-2.

D. Vasilevsky (Ex. 1008)

77. Vasilevsky, entitled “Centralized Digital Video Recording System with Bookmarking and Playback from Multiple Locations,” is generally directed to a centralized DVR system (centralized media system) located in a home with front-end receiver/decoding circuitry that receives television programming, decodes it, and stores it in a shared home DVR memory 118, which can be recalled for playback to user receivers 124-132. The users in the home can create bookmarks which are stored in a shared bookmark database 162 that is available to receiver devices 124-132. The centralized system of Vasilevsky is hosted on a home area network (HAN). Vasilevsky, Abstract, Fig. 1.

E. Logan (Ex. 1009)

78. Logan, entitled “Audio and Video Program Recording, Editing and Playback System Using Metadata,” is generally directed to a system in which television programming is parsed into segments, each of which can be associated with metadata (bookmarks). The segments and metadata can be stored in a remote facility on a public network or stored at a user location. Logan, Abstract, Figs. 1-2.

VIII. CLAIMS 13-15 AND 17-20 ARE NOT ANTICIPATED BY GUPTA (GROUND 1)

A. Claims 13-15 and 17-20 Are Patentable Over Gupta

79. Gupta fails to anticipate at least the following limitations of Claim 13:

- “*providing a computer system having at least one computer readable medium comprising a video editing environment, and being disposed at a server side of a public network*” (Limitation 13.1);
- “*receiving a set of video information at the video editing environment*” (limitation 13.2);
- “*receiving a set of cue point information at the video editing environment*” (limitation 13.3);
- “*receiving a cue point editing command over the network from a remote client terminal ...*” (limitation 13.4); and
- “*modifying a portion of the cue point information at the server side of the public network according to the cue point editing command received ... from the client terminal*” (imitation 13.5).

B. Gupta Fails to Anticipate Claim 13 of the '286 Patent

80. As discussed above in Section V.B, during prosecution the inventor repeatedly confirmed that Claim 13 provides that the *video editing environment* corresponds to a server-side application(s) that allows the client to perform remote cue point editing, as opposed to local cue point editing using a local client application.

C. The Ordinary and Customary Meaning of Limitation 13.1 In View of the Intrinsic Evidence

81. The ordinary and customary meaning of Limitation 13.1 is consistent with the claim language, the prosecution history, and the specification of the '286 Patent.

82. A POSITA would understand “*computer system ... comprising a video editing environment, and being disposed at a server side of a public network*” as referring to a server-side video editing environment with one or more applications for performing editing of cue point information to enable remote cue point editing by the client terminal as further recited in the claim. Further, the prosecution history and the specification confirm that the server-side “*video editing environment*” includes an application(s) enabling remote cue point editing by the client terminal, as opposed to local cue point editing using a client-side application as described in the Background of the Invention. See '286 Patent, 2:16-25.

1. Prosecution History

83. During the course of examination, the inventor repeatedly emphasized that the claimed invention distinguished the prior art by providing a decentralized architecture with a *video editing environment* comprising a server-side application(s) that allows the client terminal to remotely edit cue points over a public network, thus providing for **remote cue point editing** by the client as opposed to **local cue point editing** using a local client application. See Ex. 1010 ('286 Patent File History), 111-113. For example, the inventor stated: “[T]he seller can ... create and modify those

cue points **remotely**, via a remote video [editing] environment **running on an application server ...**” *Id.*, 112. The inventor also stated: “Accordingly, even if, arguendo, Shibata could be construed as teaching the cue point handling functionality recited in the claims, it would apparently only teach **local cue point handling ...** at the client terminal performing the editing. This is different from **remote editing** of and/or interaction with cue points associated with a video file **over a public client-server architecture.**” Ex. 1010 (’286 Patent File History), 112. The inventor stated: “None of the cited art, alone or in combination, apparently teaches or suggests **remote editing** of and interaction with cue points associated with a video file **over a public client-server architecture.**” Ex. 1010 (’286 Patent File History), 111. The inventor stated: “In particular, both [prior art] implementations clearly describe **local editing functionality**, and only appear to use the network for distribution of content. For example, even though the plug-in application communicates with a host, it is still installed locally as a client plug-in to facilitate **local editing** and playback functions. ... The combination would still seem to fail to teach or suggest the type of **remote cue point handling** recited in the claims.” Ex. 1010 (’286 Patent File History), 113.

84. Accordingly, the prosecution history supports the understanding of limitation 13.1 as referring to a server-side video editing environment application(s) for editing cue points to enable remote cue point editing at the client, as opposed to

local cue point editing relying on a local client-side application. This consistent characterization of the claim language during prosecution informs the proper meaning of Limitation 13.1.

85. I note that the institution decision stated: “First, though not entirely clear, Patent Owner may be reading some type of application program function into the recite claim language ‘*video editing environment*.’” Instn. Decl., 23. The claimed “*video editing environment*” is server-side software that performs the editing of cue point information. That understanding is supported by the claim language, the specification, and prosecution history discussed in this Section VIII.C.

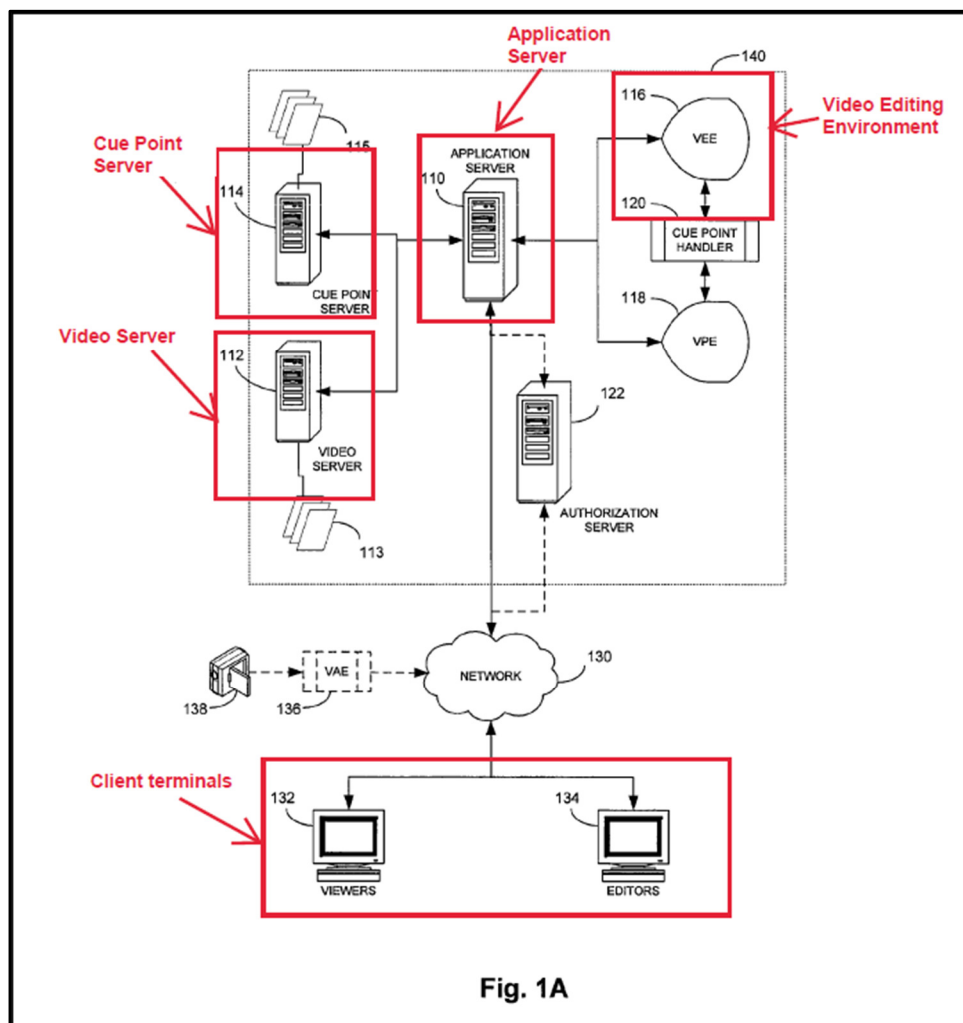
2. Claim Language

86. The relationship between the claim limitations supports the plain and ordinary meaning of Limitation 13.1 in the context of the ’286 Patent. Limitation 13.1 provides for a “*computer system ... comprising a video editing environment [VEE] ... at a server side of a public network*.” Limitations 13.2-13.3 then provide for “*receiving a set of video information at the [server-side] VEE*” and “*receiving a set of cue point information at the [server-side] VEE*.” Limitations 13.4-13.5 then provide for “*receiving a cue point editing command over the network from a remote client terminal*” and “*modifying a portion of the cue point information at the server side ... according to the cue point editing command*.” ’286 Patent, Cl. 13.

87. A POSITA would understand that the limitations call for a server-side *video editing environment* application for editing cue points to enable remote cue point editing by the client which sends a command to initiate the editing at the server-side *video editing environment* application.

3. Disclosure in Specification of the '286 Patent

88. As discussed in Section V.B, the '286 Patent consistently describes a server-side *video editing environment* application(s) that enables remote cue point editing by client terminals, as described in the figure below and associated citations:



'286 Patent, Fig. 1A (annotated in red).

89. In the '286 Patent, the *video editing environment* is consistently described as corresponding to one or more server-side applications executed by the application server to permit client devices to remotely edit cue point information associated with video data. '286 Patent, 4:1-2 (“the video editing environment comprises a server-level video editor”), 9:19-25 (“application server 110 may be configured to handle **various applications, including a Video Editing Environment (VEE)** 116 ... **These programs** may be stored on the application server 110 ... able to be executed ... by the application server 110”), 13:2-6 (“the VEE ... load[s] as a ... **server-level application**”), Figs. 1A-1B, 3A-3C. “[T]he cue point information form may be displayed on a client terminal while the cue point editor resides on a server terminal.” '286, 23:18-20. This repeated characterization of the *VEE* confirms the ordinary and customary meaning I have described.

90. The '286 Patent references an alternative embodiment where the VEE is a client-side application. '286 Patent, 13:2-4 (“VEE may load ... as a client-level application”), 13:64-65 (“with the VEE itself running either at the client- or server-level”). This is an unclaimed embodiment because the claim language and prosecution history are clear that the VEE is a server-side application.

91. The Petitioner’s expert confirms this in stating that the system has “**application server 110 executing a Video Editing Environment (‘VEE’).**” Ex.

1002 (Houh Decl.), ¶ 47 (emph. added). This confirms that the VEE comprises a server-side application for editing cue points.

92. In summary, the repeated description in the '286 Patent supports the understanding of limitation 13.1 (and limitations 13.2-13.5) as referring to a server-side *video editing environment* application that performs editing of cue points to enable remote cue point editing by the client terminal.

D. Gupta Fails to Disclose *Providing a Computer System ... Comprising a Video Editing Environment, and Being Disposed at a Server Side of a Public Network Per Limitation 13.1*

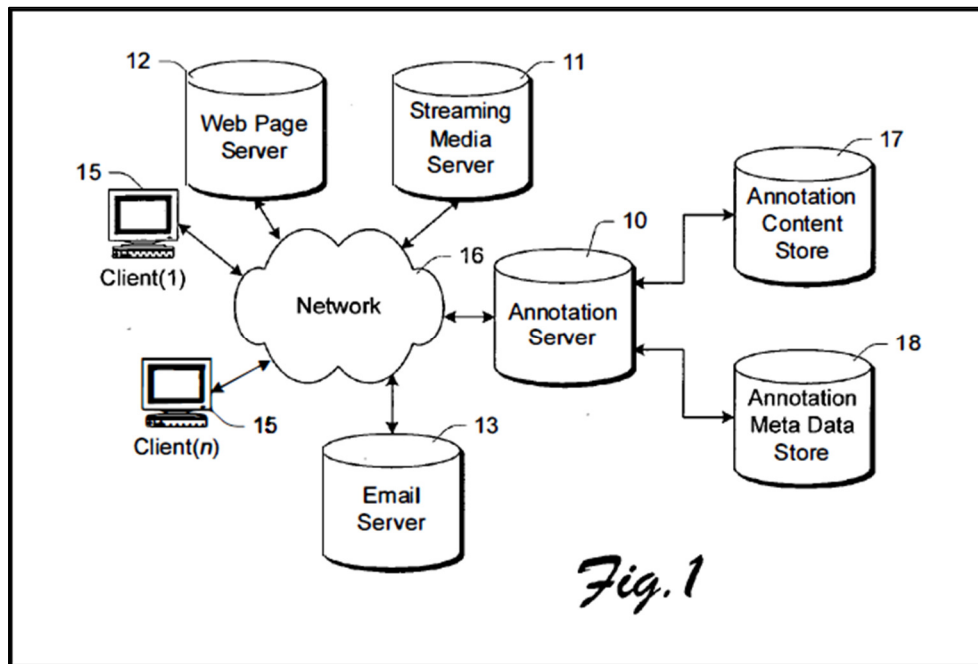
93. As set forth in limitation 13.1 (in conjunction with limitations 13.2-13.5), Claim 13 requires a *video editing environment* application that runs on the server-side of a public network and thus enables remote cue point editing by a client terminal. I note that limitations 13.4-13.5 further provide that the editing of cue point information occurs on the server-side.

94. A POSITA would understand that Gupta does not disclose a server-side *video editing environment* application that enables the remote editing of cue points by a client terminal, as required by limitations 13.1-13.5. If Gupta discloses a video editing environment, it is implemented on the client-side at client 15.

1. Gupta Discloses a *Client-Side* User Interface Module 152 that Performs *Client-Side* Cue Point Editing

95. Gupta discloses a system for creating annotations associated with media content. For example, Figure 1 of Gupta describes a system that includes

clients 15, a network (Internet) 16, streaming media server 11, web page server 12, and annotation server 10 coupled to backend databases including an annotation content store 17 and an annotation metadata store 18:



Gupta, Fig. 1.

96. Gupta describes that client computers 15 interact with servers 10-13 in Figure 1 to receive annotations associated with requested video as follows (all emph. added):

[0029] FIG. 1 shows a client/server network system and environment in accordance with the invention. Generally, the system includes multiple network server computers 10, 11, 12, and 13, and multiple (n) network client computers 15. The computers communicate with each other over a data communications network. The communications network in FIG. 1 comprises a public network 16 such as the Internet.

The data communications network might also include, either in addition to or in place of the Internet, local-area networks and/or private wide-area networks.

[0030] Streaming media server computer 11 has access to streaming media content in the form of different media streams. These media streams can be individual media streams (e.g., audio, video, graphical, etc.), or alternatively composite media streams including two or more of such individual streams. ...

[0032] Annotation server 10 controls the **storage of annotations** and **their provision to client computers 15**. The annotation server 10 manages the annotation meta data store 18 and the annotation content store 17. **The annotation server 10 communicates with the client computers 15 via any of a wide variety of known protocols, such as the Hypertext Transfer Protocol (HTTP). The annotation server 10 can receive and provide annotations via direct contact with a client computer 15,** or alternatively via electronic mail (email) via email server 13. The annotation server 10 similarly communicates with the email server 13 via any of a wide variety of known protocols, such as the Simple Mail Transfer Protocol (SMTP).

[0033] The annotations managed by annotation server 10 correspond to the streaming media available from media server computer 11. In the discussions to follow, the annotations are discussed as corresponding to streaming media. However, it should be noted that the annotations can similarly correspond to “pre-delivered” rather than streaming media, such as media previously stored at the client computers 15 via the network 16, via removable magnetic or optical disks, etc.

[0034] When a user of a client computer 15 accesses a web page containing streaming media, a conventional web browser of the client computer 15 contacts the web server 12 to request a Hypertext Markup Language (HTML) page. The client-based browser also submits requests to the media server 11 for streaming data, and the annotation server 10 for any annotations associated with the streaming data. When a user of a client computer 15 desires to **add or retrieve annotations**, the client computer 15 contacts the annotation server 10 to **perform the desired addition/retrieval**.

Gupta, [0029]-[0030], [0032]-[0034].

97. At the very outset in the Summary of the Invention, Gupta discloses that annotation editing (cue point editing) is performed client-side, including creating annotation parameters such as temporal range selection, annotation creation, and annotation starting point identification:

[0009] In a networked client/server system, media content as well as annotations corresponding to the media content can be transmitted from a server(s) to a client. A user interface is presented to the user to facilitate creating new annotations and viewing annotations.

[0010] According to one aspect of the invention, annotations correspond to a temporal range of the media content as defined by a temporal beginning point and a temporal ending point. **When a user desires to create a new annotation the client assists the user in determining the temporal range of the media content** to which the new annotation is to correspond. **The client receives the user request to create the new annotation**, which often times is made after the

desired temporal beginning point for the new annotation has passed. **The client analyzes the media content and detects likely beginning points for the new annotation and identifies these beginning points to the user.** The user can then select one as the beginning point for the new annotation. Analogous assistance can be provided in determining the ending point for the new annotation.

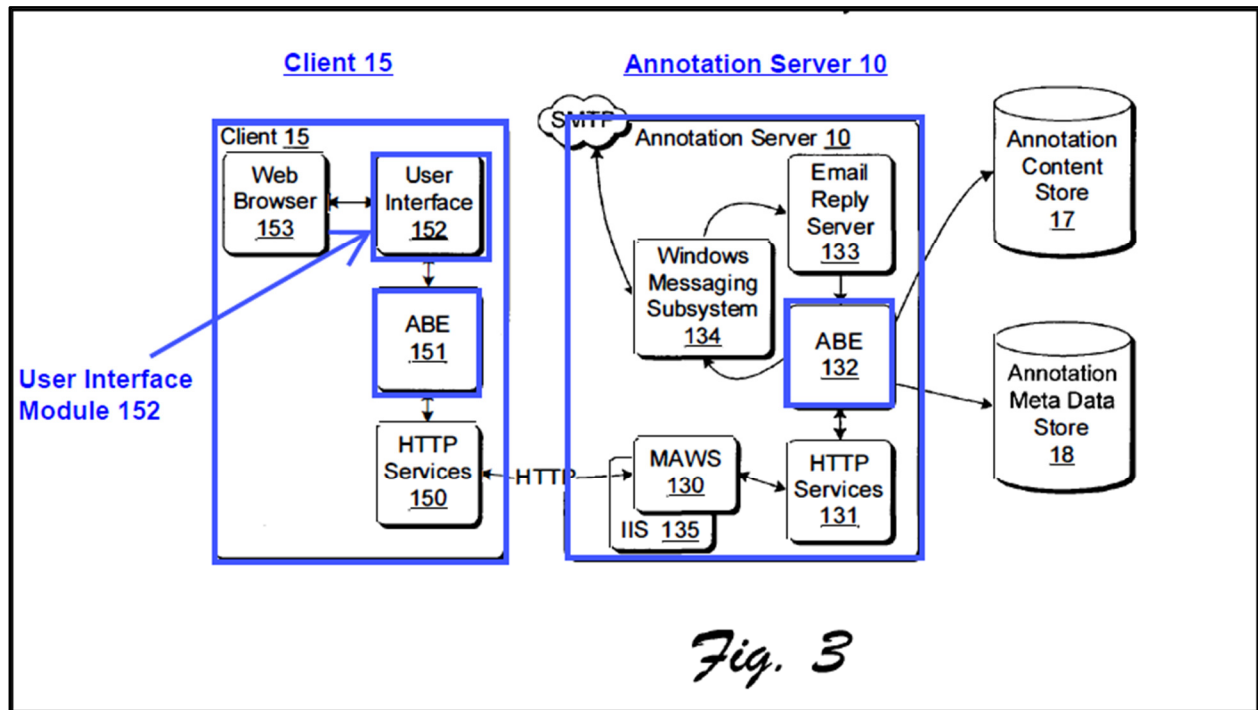
Gupta, [0009]-[0010], Fig. 5.

98. Gupta is clear that annotation server provides for the server-side storage and delivery of annotations, not server-side editing annotations as required by limitation 13.1: “Annotation server 10 controls the **storage** of annotations and their **provision** to client computers 15.” Gupta, [0032]. If Gupta’s annotation server had server-side software for editing annotations, that passage would have read something like: “Annotation server 10 controls the storage of annotations [, **the editing of annotations**,] and their provision to client computers 15.” The absence of such language indicates that Gupta’s annotation server does not run server-side software for editing annotations.

a. *Client-Side* UIM 152 Application, Not Annotation Server 10, Creates and Edits Annotations in Gupta

99. As noted above, Gupta discloses that the server-side functionality of annotation server is to store annotations in the database and deliver annotations to clients, not to edit annotations: “Annotation server 10 controls the **storage** of annotations and their **provision** to client computers 15.” Gupta, [0032].

100. Figure 3 of Gupta illustrates the client 15 (and its internal components) and the annotation server 10 (and its internal components) coupled to data stores 17 and 18 that store annotation information:

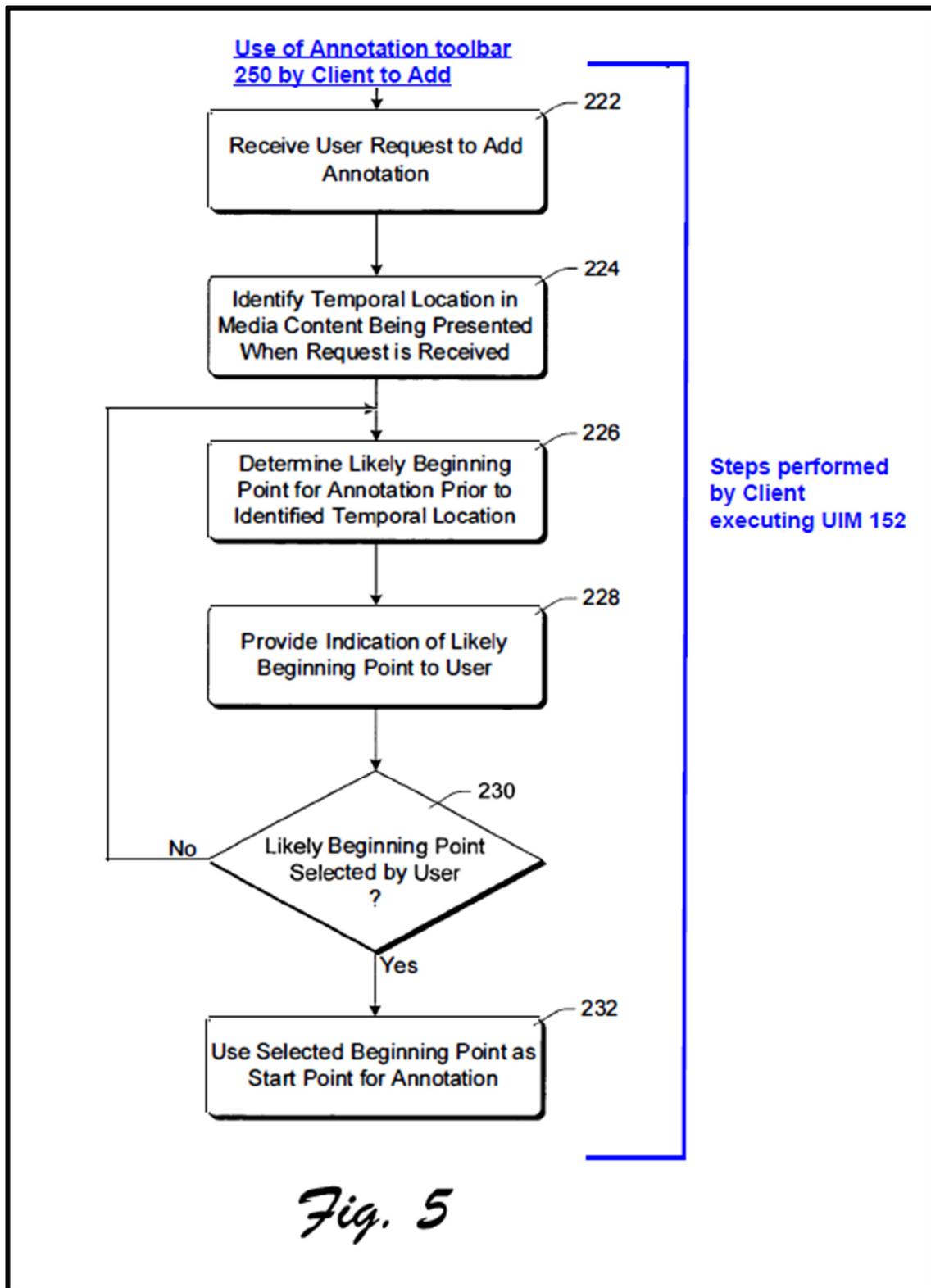


Gupta, Fig. 3 (annotated in blue).

101. Gupta discloses **local cue point editing** performed by client 15 executing a **client-side application (UIM 152)**, as opposed to remote cue point editing carried out by the annotation server 10 (and/or other servers 11-13) executing a server-side *video editing environment* application. With reference to Figure 3 above, Gupta confirms that annotations (cue points) are created and edited at client 15, not the annotation server, by stating: “The annotation server 10 can receive ... annotations by direct contact **with a client computer 15.**” Gupta, [0032]. If the

annotation server is receiving annotations from the client computer, a POSITA would understand that to mean that it is the client computer that creates the annotations. Those annotations are created (and edited) using **the local client application** user interface module (UIM) 152, which **runs on client device 15** and “provides the user interface (UI) for a user to **add ... annotations**, and be presented with the annotations.” Gupta, [0045], [0082] (“The [user] interface module 152 is an **application program ... that is executed at client 15.**”), Fig. 3 (UIM 152). “**Annotation content authored at client 15, using server interface 152,** is **received** by [the server-side] ABE 132 and maintained in annotation store 17.” [0048]. That language plainly describes the client executing UIM 152 to create new annotations on the client-side, which are transmitted to the annotation server for storage. For example, “authored” indicates that it is client 15 executing the UIM application, not annotation server 10, that **creates** the annotations.

102. Figure 5 of Gupta discloses a process for creating/editing annotations that “is implemented by [the client-side] user interface module 152 of FIG. 3”. Gupta, [0075], Fig. 5. The steps described in Figure 5 for adding an annotation are performed at the client 15, not at annotation server 10.



Gupta, Fig. 5 (annotated), [0075].

103. The above-cited passages from Gupta are reproduced below (all emph. added):

[0032] Annotation server 10 controls the storage of annotations and their provision to client computers 15. The annotation server 10 manages the annotation meta data store 18 and the annotation content store 17. **The annotation server 10 communicates with the client computers 15 via any of a wide variety of known protocols, such as the Hypertext Transfer Protocol (HTTP). The annotation server 10 can receive and provide annotations via direct contact with a client computer 15 ...**

[0045] Client 15 runs an HTTP services (HttpSvcs) module 150, which manages communications with server 10, and an annotation back end (ABE) module 151, which translates user actions into commands destined for server 10. **A user interface (MMA) module 152 provides the user interface (UI) for a user to add and select different annotations, and be presented with the annotations.**

[0048] Annotation server 10 further includes an annotation back end (ABE) module 132, which contains functionality for accessing annotation stores 17 and 18, for composing outgoing email based on annotation data, and for processing incoming email. Incoming email is received and passed to the ABE module 132 by the Email Reply Server 133. **Annotation content authored at client 15, using user interface 152, is received by [server-side] ABE 132 and maintained in annotation content store 17.** Received meta data (control information) corresponding to the annotation content is maintained in annotation

meta data store 18. The annotation content and meta data can be stored in any of a variety of conventional manners, such as in SQL relational databases (e.g., using Microsoft “SQL Server” version 7.0, available from Microsoft Corporation). **Annotation server 10 is illustrated in FIG. 3 as maintaining the annotation content and associated control information (meta data) separately in two different stores.**

[0075] FIG. 5 is a flowchart illustrating an exemplary process for assisting a user in determining a starting point for an annotation. **The process shown in FIG. 5 is implemented by user interface module 152 of FIG. 3, and may be performed in software. These steps are described with additional reference to FIG. 3.**

[0081] **An annotation can be created by a user of any of the client computers 15 of FIG. 1.** As discussed above with reference to FIG. 3, client 15 includes a[] [user] **interface module 152 that presents an interface to a user (e.g., a graphical user interface (GUI)), allowing a user to make requests of annotation server 10. Numerous windows, dialog boxes, tool bars, etc. can be displayed to the user by interface module 152, as discussed in more detail below. In the illustrated embodiment, a user can access annotation server 10 via an annotation toolbar provided by interface 152.**

[0082] **The interface module 152 is an application program (or part of an application program) that is executed at client 15.**

Gupta, [0032], [0045], [0075], [0048], [0081]-[0082] (all emph. and underscore added).

104. The Board (DI, 25) referenced paragraph [0049] of Gupta, which states:

[0049] Each of the annotations **maintained in annotation stores 17 and 18** corresponds to each of the different versions of particular multimedia content available to media server 11. Thus, regardless of the number of different versions of particular multimedia content available to media server 11, **each annotation created by annotation server 10 is maintained as a single copy** corresponding to all of these different versions.

Gupta, [0049] (emph. added).

105. The question raised by paragraph [0049] is whether “created by annotation server 10” means the annotation server 10 is creating new annotations or instead that annotation server 10 is “creating” annotations by storing new annotations, created by the client computer 15, on annotation stores 17 and 18. First, the function of annotation server 10 is explicitly described as storing and retrieving annotations. Gupta, [0032] (“Annotation server 10 controls the storage of annotations and their provision to client computers 15.”). Nowhere else in Gupta is there any suggestion that annotation server 10 creates annotations. See Gupta, [0032]. Second, a POSITA would understand the sentence in paragraph [0049] as referring to annotations being maintained by annotation server 10 in data stores 17 and 18, not that annotations are created at annotation server 10. A POSITA must examine Gupta as a whole, and in doing so would understand that it discloses that annotations are created and edited at the client-side UIM 152 module, not the annotation server 10.

106. Read as a whole, Gupta clearly discloses local annotation editing performed using local client UIM 152 running on the client 15, as explained above in multiple passages and figures. See Gupta, [0032], [0045], [0048], [0081]-[0082], Fig. 3, Fig. 5 (all explained in the paragraphs above), Fig. 15, Fig. 21 (explained in Section VIII.D.1.b below).

107. The Board suggested that Gupta's UIM 152 module "is similar to the user interface" of the '286 Patent. DI, 24. I respectfully disagree. A POSITA would understand this not to be the case because in Gupta the "[user] interface module 152 is an **application program ... that is executed at client 15.**" Gupta, [0082] (emph. added), Fig. 3. And as described in this Section VIII.D.2 and Section VIII.D.3, Gupta's UIM 152 module is a client-side application program that provides for **local client editing of annotations** by generating dialog boxes for the user to **locally create** new annotations (Figure 7) and to **locally edit** annotations (Figure 21). In contrast, the user interface disclosed and claimed in the '286 Patent is an interface to a *video editing environment* application at the server-side supporting remote cue point editing (e.g., remote annotation editing). That is different from the user interface of Gupta's UIM 152, which does not interface with a server-side *video editing environment* application as claimed. As previously explained (and explained in further detail in Sections VIII.D.3-5 below), there is no server-side *video editing*

environment application for performing editing of annotations in Gupta corresponding to limitation 13.1.

108. A POSITA would understand the above passages as demonstrating that Gupta discloses local cue point editing by client 15 executing client-side application UIM 152. Gupta does not disclose remote cue point editing carried out by the annotation server 10 executing a server-side *video editing environment* application for editing annotations as required by limitation 13.1, much less such an application that causes server-side cue point editing per limitations 13.4-13.5.

b. *Client-Side* UIM 152 Generates the Client-Side Dialog Boxes Including the “Edit Annotation” Figure 21 Used to Edit an Annotation

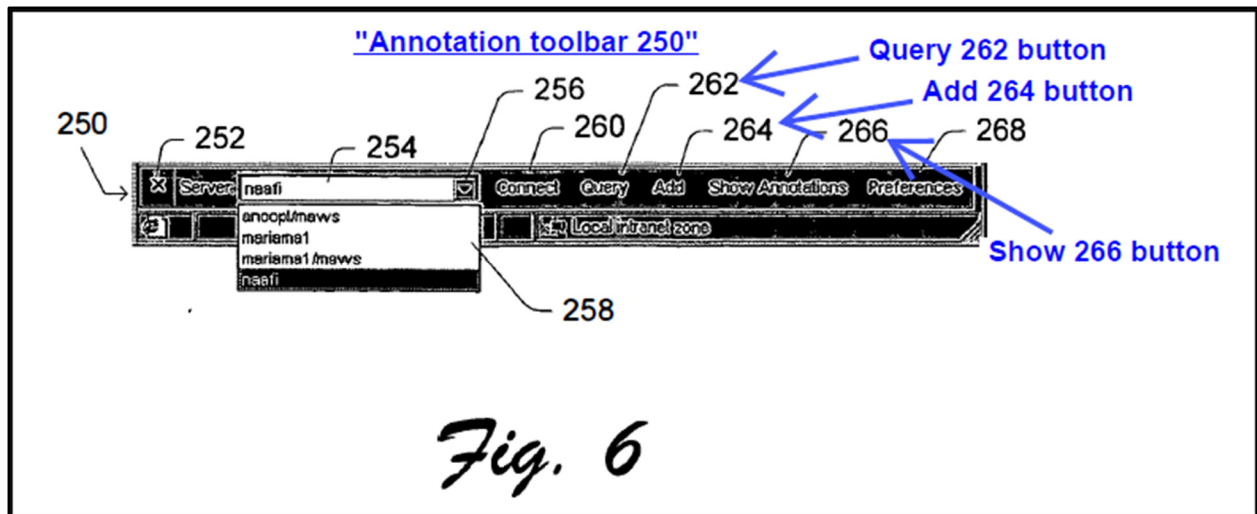
109. Gupta describes how the client-side UIM 152 provides “[n]umerous windows, **dialog boxes**, toolbars, etc. [to] be displayed to the user by interface module 152, as discussed in more detail below.” [0081] (emph. added). A POSITA would understand this to mean that the client-side UIM 152 generates the dialog boxes that allow the user to locally create and edit annotations (cue points), as later described in Gupta’s specification, including the user interfaces of Figures 6-21.

110. It is well known in the art that a dialog box is a client-side function provided at the client-side browser application. Ex. 2036 (Seiko), [0109], [0111]; Ex. 2037 (IBM), [0034]; Ex. 2038 (Cisco), 2:14-21.

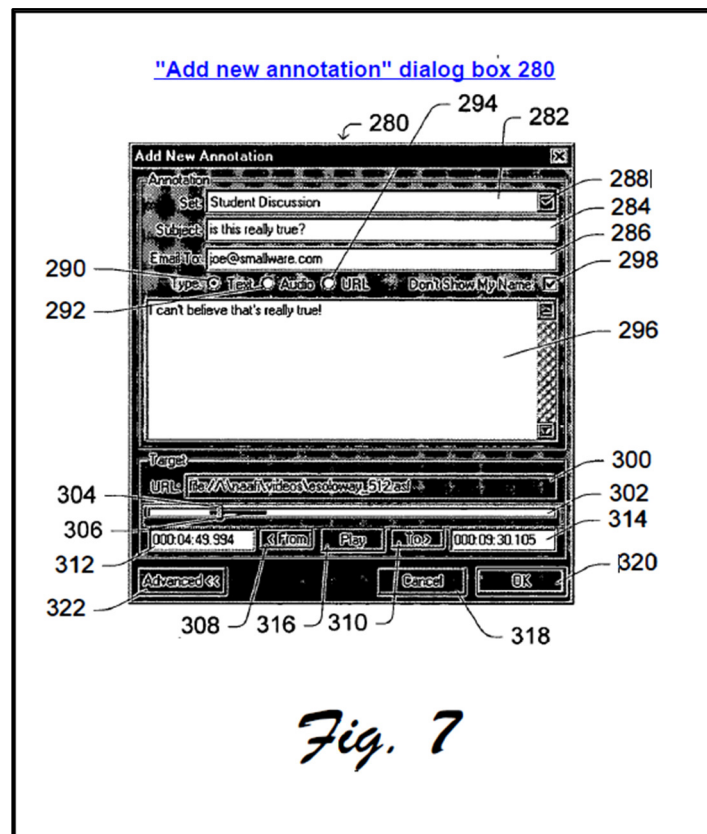
111. Gupta discloses that client-side UIM 152 generates a number of client-side dialog boxes. First, Gupta describes that the client-side UIM 152 generates the “Annotation toolbar” 250 of Figure 6 which provides various functions for creating and editing annotations. The Annotation toolbar has buttons including an “add annotation” 264, which can be used by UIM 152 to generate the “Add annotation” dialog box of Figure 7 used to create a new annotation. Gupta, [0086], [0089], Figs. 6-7. The Annotation toolbar of Fig. 6 also has a “query” 262 tab, which can be selected to provide the “Query annotations” dialog box of Figure 14. The Query annotations dialog box has an “annotation set list” 426 that can be selected to provide the “View annotations” dialog box of Figure 15. Gupta, [0086], [0120]-[0124], Figs. 6, 14, 15. The user can also select the “show annotation” 266 tab in the Annotation toolbar 250 of Figure 6 to directly transition to the View annotations dialog box of Figure 15. See Gupta [0084]-[0088], Figs. 6, 7, 14, 15.

112. All of these client-side dialog boxes including those for creating annotations (Fig. 7) and editing annotations (Fig. 21) are generated by the client-side UIM 152 application, as disclosed in the above-cited passages and figures of Gupta.

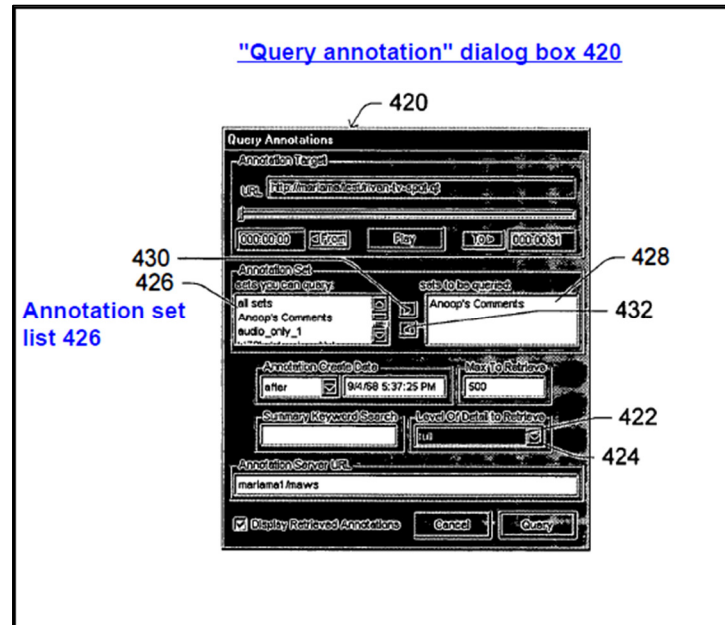
113. Figures 6, 7, 14, and 15 of Gupta are depicted below:



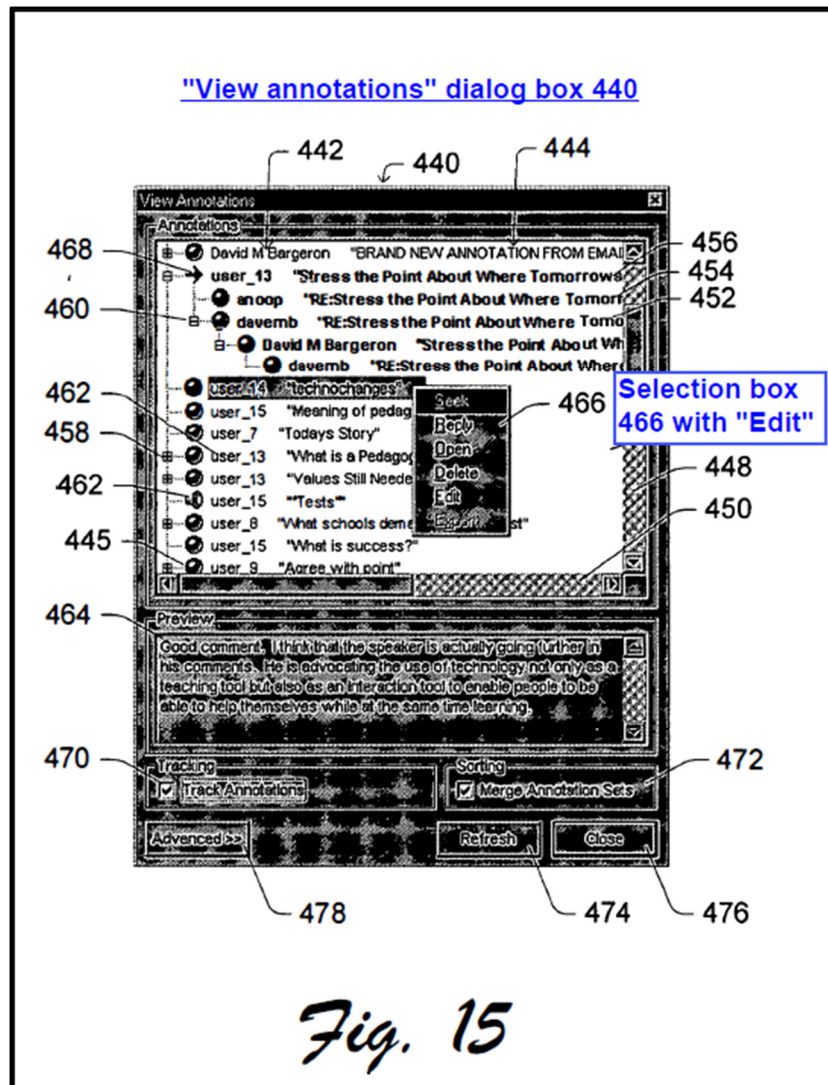
Gupta, Fig. 6 ("Annotation toolbar" generated by UIM 152 on client 15, annotated in blue, see [0081]-[0086]).



Gupta, Fig. 7 (“Add annotation” dialog box 280, generated by UIM 152 on client 15, annotated in blue, see [0081]-[0082], [0086]-[0101].



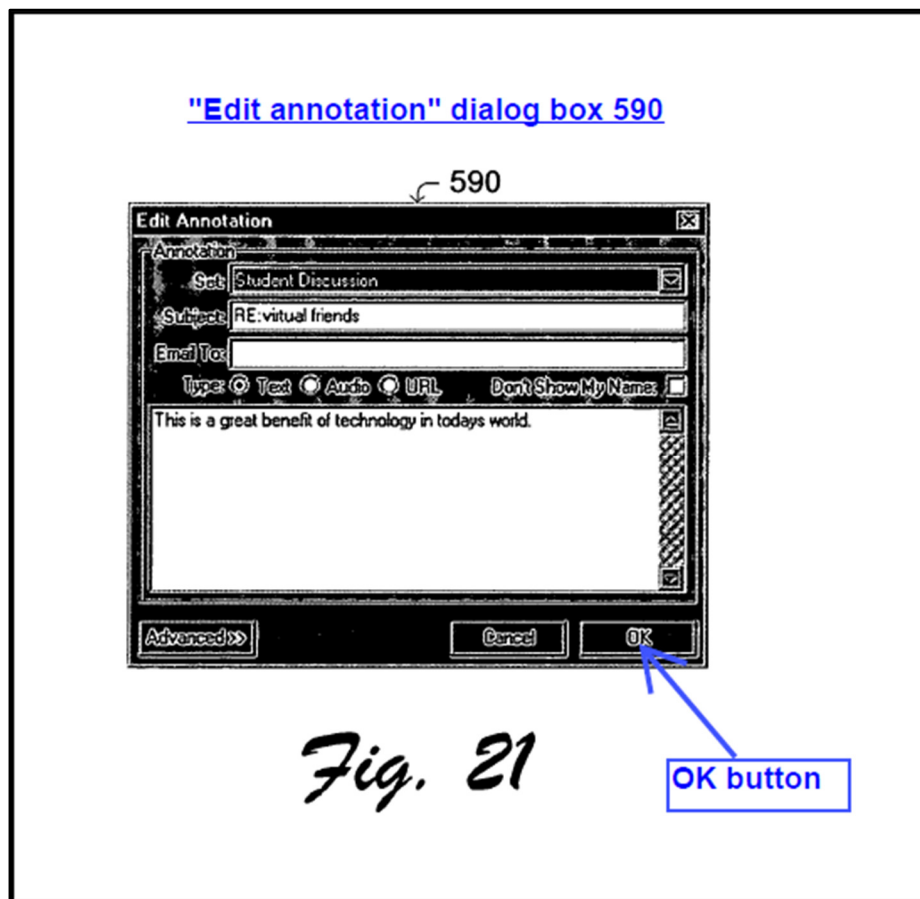
Gupta, Fig. 14 (“Query annotation” dialog box 420, generated by UIM 152 on client 15, annotated in blue, see [0081]-[0082], [0120]-[0123].



Gupta, Fig. 15 (“View annotations” dialog box 440, generated by UIM 152 on client 15, annotated in blue, including selection box 466 with “edit” option). See Gupta, [0081]-[0082], [0087], [0124]-[0132].

114. In the View annotations dialog box of Figure 15 (depicted above), the client-side UIM 152 provides selection box 466 with the “edit” tab that allows the user to access the “Edit annotation” dialog box of Figure 21. Figure 21 allows for the editing of existing annotations, which are transmitted to the annotation server 10

upon selection of the OK button. Gupta, [0124]-[0128], [0144], [0100] (OK button), Figs. 15, 21. Gupta states: “Selection of the ‘edit’ feature [in the View annotation dialog box of Fig. 15] **causes [client-side user] interface module 152** to display another dialog box [Edit annotations dialog box of Fig. 21] **allowing the user to edit the content of the annotation.**” Gupta, [0128] (emph. added), Figs. 15, 21. Figure 21 is depicted below:



Gupta, Fig. 21 (“Edit annotation” dialog box 590, generated by UIM 152 on client 15, annotated in blue, see [0081]-[0082], [0124]-[0128], [0144], [0100]).

115. Paragraph [100] describes the operation of the “OK” button for Figure 7 (local Add annotation dialog box), and it also applies to Figure 21 (local Edit annotation dialog box). Paragraph [0100] states: “Selection of OK button 320 causes **[client-side user] interface [module] 152 to forward the information entered in [the] dialog box []** to annotation backend 151, which sends **the new annotation information** [entered into the locally-generated Edit annotation dialog box of Figure 21] to annotation server 10 of FIG. 3.” Gupta, [0100] (emph. added). This statement, like many others, confirms that the annotations in Gupta are locally edited by the client-side UIM 152, which then forwards the information to the annotation server 10 for storage. Gupta does not disclose that annotation server 10 is performing any server-side editing of the annotation, as required by the server-side *video editing environment* application of Limitation 13.1. Accordingly, the Petitioner’s argument (at 17) based on the OK button does not establish Gupta as disclosing the server-side *video editing environment* application of Limitation 13.1.

116. The passages pertaining to the Edit annotation dialog box (Fig. 21) relied upon by the Petitioner demonstrate that it is generated by the client-side UIM 152, not the server-side annotation server 10:

[0124] FIG. 15 shows a dialog box 440 that identifies annotations corresponding to one or more annotation sets. The identified annotations are the result of the query input by the user as discussed above with reference to FIGS. 12-14, or alternatively by selection of

show annotations button 266 of FIG. 6. In the illustration of FIG. 15, annotation identifiers in [the] form of user identifiers 442 and subject lines 444 are displayed within an **annotation listing box 446**. The user can scroll through annotation identifiers in a conventional manner via scroll bars 448 and 450. ... Icons identifying the type of annotation are also **displayed in listing box 446**, such as a circle 445 to represent text and a microphone 447 to represent audio.

[0125] Related annotations are displayed in an **annotation listing 446** in a hierarchical, horizontally offset manner. The identifier of an annotation that is related to a previous annotation is “indented” from that previous annotation’s identifier and a connecting line between the two identifiers is shown. ...

[0126] By **selecting a particular annotation identifier from annotation listing 446** (e.g., “single clicking” on the summary using the left mouse button), preview information is presented in a preview section 464, and a selection box or menu 466 is provided. The exact nature of the preview information is dependent on the amount of information that was requested (e.g., as identified in level of detail 422 of FIG. 14).

[0127] **Selection box 466** provides various user-selectable features related to the annotation whose identifier is selected, including “seek”, “reply”, open “delete”, and “**edit**”. Selection of the “seek” feature causes interface module 152 of FIG. 3 to initiate presentation of the media segment corresponding to the annotation (analogous to the play button 316 of FIG. 7 discussed above). Selection of the “reply” feature

allows the user to create a new annotation that is related to the selected annotation. The new annotation will be displayed offset horizontally from the selected annotation, analogous to annotations 452 and 456 discussed above.

[0128] Selection of the “open” feature causes interface module 152 of FIG. 3 to request all of the annotation content for the selected annotation from annotation server 10. Selection of the “delete” feature causes interface module 152 to remove the selected annotation from annotation listing 446. In one embodiment, removal of a particular annotation also causes removal of all related annotations (e.g., deleting the annotation identified by identifier 456 would also cause deletion of the annotations identified by identifiers 452 and 454). **Selection of the “edit” feature causes interface module 152 to display another dialog box [Fig. 21] allowing the user to edit the content of the annotation, as discussed in more detail below. ...**

[0144] **FIG. 21 illustrates an exemplary “edit annotation” dialog box 590. Edit annotation dialog box 590 is displayed in response to a user request to edit an annotation, such as selection of the “edit” option in menu 466 of FIG. 15.** Dialog box 590 includes information and fields displaying the current information for the annotation, analogous to dialog box 560 of FIG. 20. New information can be entered into the fields, analogous to add annotation dialog box 330 of FIG. 8. In the illustrated example, existing email addresses cannot be altered and new email addresses cannot be entered.

[0100] Dialog box 280 also includes a cancel button 318, **an OK button 320**, and an advanced button 322. Selection of cancel button

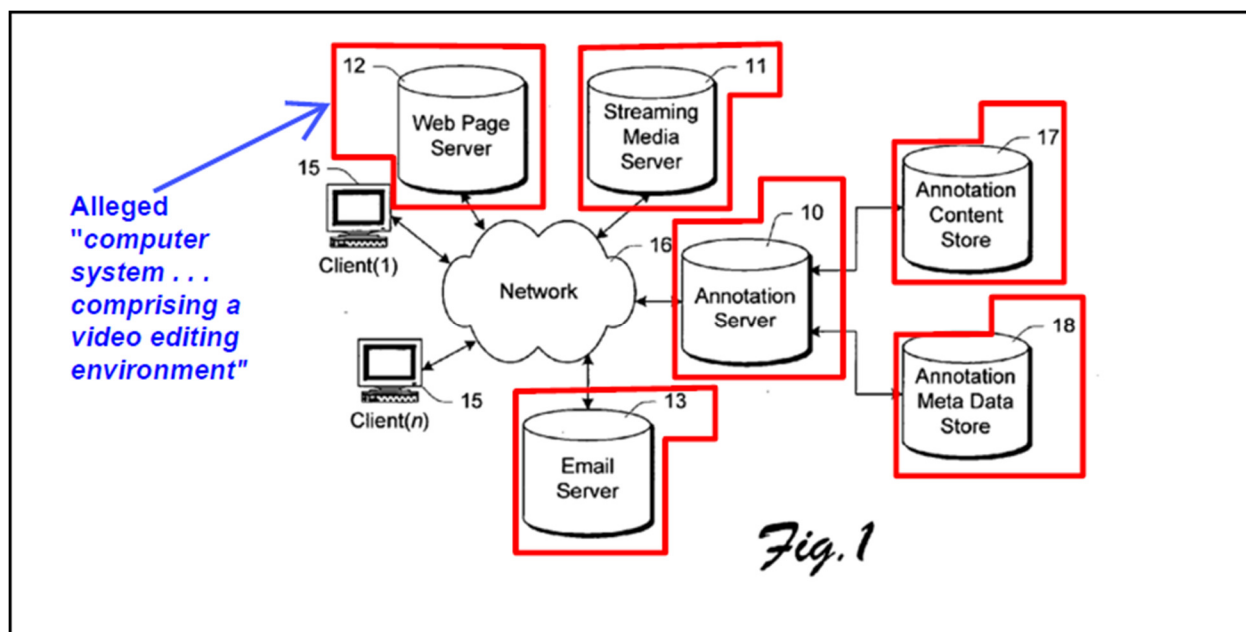
318 causes interface 152 to close the dialog box 280, canceling out of the add new annotation process. **Selection of OK button 320 causes [client-side user] interface 152 to forward the information entered in dialog box 280 to annotation backend 151, which sends the new annotation information to annotation server 10 of FIG. 3.** Interface 152 then closes dialog box 280, the new annotation being entered.

Gupta, [0124]-[0128], [0144], [0100].

117. As explained above, the Petitioner's reliance on the dialog boxes of Figures 15 and 21 for limitation 13.1 (Pet., 16-17) is off point. A POSITA would understand the above passages as demonstrating that Gupta discloses local cue point editing by client 15 executing the **client-side application UIM 152** to provide those dialog boxes that are used to locally edit annotations. Gupta does not disclose remote cue point editing performed by the annotation server 10 (and/or other servers 11-13) running a server-side *video editing environment* application per Limitation 13.1.

2. Petitioner Aylo's Argument for Limitation 13.1

118. The Petitioner and its expert assert that the claimed server-side *computer system ... comprising a video editing environment* application is disclosed by Gupta's annotation server 10, coupled with three other separate servers 11-13 and databases 17-18, as illustrated by the red-boxed components below. Pet., 16 ("The at least one computer-readable medium *comprising a video editing environment* (e.g., software for both editing annotations and streaming video, executed by servers 10-13 and stores 17-18)."); Houh Decl., ¶¶ 73, 76.



Gupta, Fig. 1 (annotated in red by Petitioner; annotated in blue by Patent Owner).

See Pet., 15.

119. But there is no disclosure in Gupta that the red-boxed components (individually or collectively) comprise a server-side *video editing environment* application for editing cue point information as claimed. Gupta does not expressly disclose that these server-side components run any particular software, let alone a server-side *video editing environment* application for editing cue points (annotations), as required. Instead, a POSITA would recognize that Gupta discloses **local cue point editing** by a client running a **client-side application (UIM 152)** on client 15, as explained above in Section VIII.D.1 and further below.

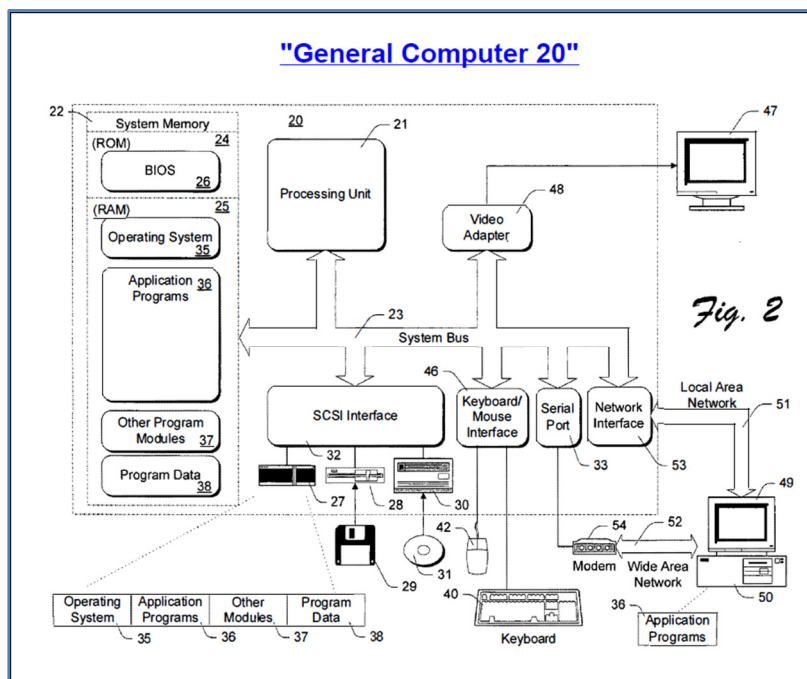
3. Gupta Does Not Disclose a Server-Side Application on Annotation Server 10 For Providing a *Video Editing Environment* per Limitation 13.1

120. A POSITA would recognize that Gupta does not disclose (see Gupta, Fig. 1, annotation server 10 a server-side *video editing environment* application as claimed. This is a point that the Petitioner concedes in later describing “the **client-side** editing functionality disclosed by Gupta.” See Pet., 72 (emph. added).

121. Gupta discloses a generic computer in Figure 2 with software programs stored in computer readable media, such as “**program modules . . . including . . . one or more application programs 36, other program modules 37, and program data 38.**” Gupta, [0040]. For purposes of institution only, and without the benefit of the Patent Owner’s expert testimony, the Board felt that Gupta might “describe[] . . . a video editing environment, such as software for both editing annotations and streaming video executed by servers 10–13 and stores 17–18.” DI, 24 (relying on Houh Decl., ¶¶ 75-76 regarding Fig. 2 of Gupta).

122. However, Gupta’s Figure 2 (which references “program modules” and “application programs 36, other program modules 37,” see [0039]-[0040]) is a generic diagram of “a **general example of a computer 20 that can be used as a client or server** in accordance with the invention. Computer 20 is shown as an example of a computer that can perform the functions of any of server computers

10-13 or a client computer 15 of FIG. 1.” Gupta, [0037]. Figure 2 for the generic computer 20 is depicted below:



Gupta, Fig. 2 (annotated in blue).

123. A POSITA would thus understand Figure 2 to describe a generic client/server with **generic program modules and application programs**. There is no disclosure whatsoever in [0037]-[0042] on Figure 2 indicating that the generic program modules/application programs are a server-side *video editing environment* application, as claimed. While Figure 2 states that the programs of generic computer 20 “can perform the functions of any [] server computer,” the remainder of Gupta’s specification never discloses that the annotation server 10 has a server-side *video editing environment* application for creating/editing annotations (cue points).

124. The standard for invalidating a claim as anticipated requires that the reference actually disclose each limitation as recited. In my opinion, the disclosure in Gupta in Figure 2 of a generic computer 20 that can be a client or a server, and that contains generic application programs, and that does not even mention annotation editing, falls far short of disclosing Limitation 13.1 for a server-side *video editing environment* application for remote cue point editing.

125. Thus, Figure 2 and Dr. Houh’s testimony regarding generic computer 20 of Figure 2 do not show that Gupta discloses a server-side *video editing environment* application running on annotation server 10 (or servers 11-13) that supports remote cue point editing by client devices per 13.1.

4. The Petitioner’s Expert’s Assertion that Annotation Server 10 Executes Software for Editing Annotations Is Contrary To the Evidence

126. Dr. Houh asserts (76) that “the **annotation server 10 executes software for editing annotations.**” Houh Decl., ¶ 76. I respectfully submit that assertion is unsupported by Gupta. Dr. Houh refers to three items in Gupta. Houh Decl., ¶ 76.

127. He refers to (1) the ABE module 132 (annotation backend, Gupta, [0048]). However, Gupta discloses that ABE 132 “contains functionality for **accessing** stores 17 and 18.” Gupta, [0048]. See Houh Decl., ¶ 76. A POSITA would understand that to mean that ABE 132 provides for storing/retrieving annotations

to/from databases 17-18. This is consistent with [0032], which states that the annotation server controls the “**storage** of annoations” [storage] and the “their **provision** to client computers” [retrieval]. Gupta, [0032]. Gupta does not disclose or suggest that ABE 132 “executes software for editing annotations,” as asserted. See Houh Decl., ¶ 76. Gupta also states: “Annotation content **authored by client 15, using [client-side] user interface [UIM] 152**, is **received** by ABE 132 and **maintained** in annotation content store 17.” Gupta, [0048]. This indicates that annoations are created at the client-side and the ABE 132 receives them from the client and stores them in the database 17. In other words, ABE 132 serves as the interface between annotation server 10 and the backend databases 17-18 for storage/retrieval, not that ABE 132 (or any other component in annotation server 10) “executes software for editing annotations,” as asserted. See Houh Decl., ¶ 76. This understanding of ABE 132 is consistent with Gupta’s statement that annotation server 10 (which contains ABE 132) functions to (1) store annotations and (2) retrieve annotations. Gupta, [0032].

128. Next, Dr. Houh (¶ 76) refers to the “edit annotation dialog box 590” of Figure 21 in support of his assertion about that annotation server 10 executes software for editing annotations. Houh Decl., ¶ 76. However, as I explained above in Section VIII.D.1.b, Figure 21 is a **client-side** dialog box generated by the **client-side** UIM 152 application. A POSITA would understand that it is not a server-side

application component that “executes software for editing annotations,” as asserted. See Houh Decl., ¶ 76.

129. Lastly, Dr. Houh refers to the selection of the “OK” button in [0100] in support of his incorrect assertion that annotation server 10 executes software for editing annotations. See Houh Decl., ¶ 76. However, as I explained above in Section VIII.D.1, the OK button is part of the dialog box (e.g., Edit annotation in Figure 21) generated by the client-side UIM 152 application, not server-side software on annotation server 10. See Gupta, [0081]-[0082]. Furthermore, Gupta discloses that the user’s selection of the OK button on the client-side UIM 152 causes “annotation backend 151 [to] send[] the new annotation information to annotation server 10 of FIG. 3.” A POSITA would understand this to disclose that the client-side edited annotation (using the client-side UIM 152 application) is **transmitted to** the annotation server 10, not that the annotation server 10 performs edits on an annotation by “execut[ing] software for editing annotations” as Dr. Houh suggests. See Houh Decl., ¶ 76.

130. Accordingly, Dr. Houh’s assertion that “the annotation server 10 executes software for editing annotations” finds no support in Gupta and is contradicted by Gupta’s disclosure of how the editing of annotations is carried out. See Section VIII.D.1-D.3.

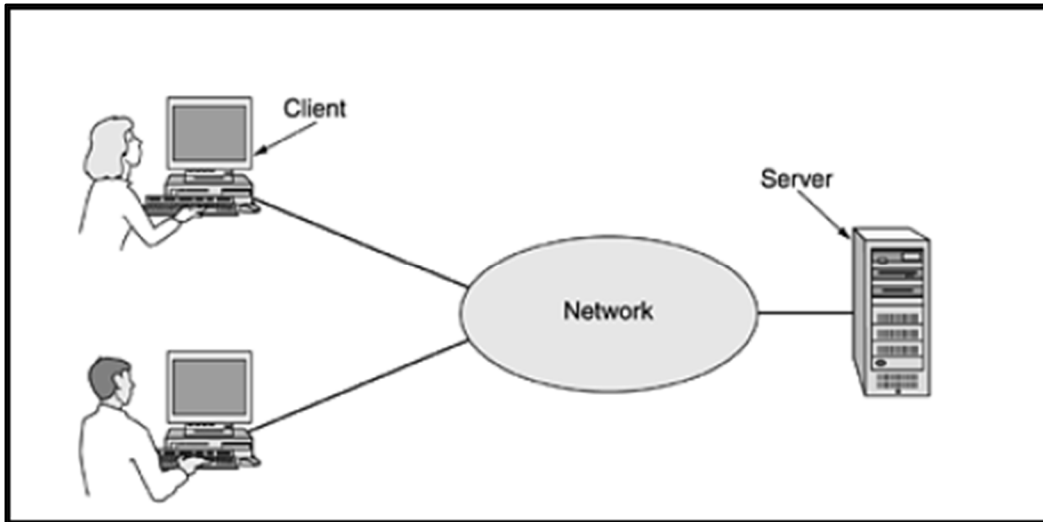
5. Gupta's Use of HTTP Requests Sent from Client 15 to Annotation Server 10 Is Standard Client-Server Communications Protocol and Does Not Mean that Annotation Server 10 is Executing a *Video Editing Environment* Application

131. In the institution decision for the -00710 IPR, the Board suggested that the client's use of ABE 151 to send commands in the form of HTTP requests is relevant to Limitation 13.1. Instn. Dec., 24-25. However, HTTP is a request-response protocol used in virtually all client-server interactions over the Internet:

HyperText Transfer Protocol (HTTP) is the method used to transfer or convey information on the World Wide Web. ... **HTTP is a request/response protocol between clients and servers.** ... A HTTP client initiates a request by establishing a Transmission Control Protocol (TCP) connection to a particular port on a remote host A HTTP server listening on that port waits for the client to send a Request Message. Upon receiving the request, the server sends back a status line, such as "HTTP/1.1 200 OK", and a message of its own, the body of which is perhaps the requested file, an error message, or some other information. Resources to be accessed by HTTP are identified using Uniform Resource Identifiers (URIs) (or, more specifically, URLs) using the http: or https: URI schemes.

Ex. 2023 (HTTP), 1.

132. The text "Computer Networks" by Tanenbaum (Ex. 2025) also describes HTTP as the standard protocol for client-server interactions over public networks like the Internet (all emph. added below):



This whole arrangement is called the client-server model. It is widely used and forms the basis of much network usage. It is applicable ... when a person at home accesses a page on the World Wide Web, the same model is employed, with the remote Web server being the server and the user's personal computer being the client. Under most conditions, one server can handle a large number of clients.

If we look at the client-server model in detail, we see that two processes are involved, one on the client machine and one on the server machine. Communication takes the form of the client process **sending a message [request] over the network to the server process. The client process then waits for a reply message [response]**. When the server process gets the request, it performs the requested work or looks up the requested data and sends back a reply [response].

Ex. 2025 (Computer Networks), 13.

The transfer protocol used throughout the World Wide Web is HTTP (HyperText Transfer Protocol). It specifies what messages clients may send to servers [requests] and what responses they get back in return [responses]. Each interaction consists of one ASCII request, followed

by one RFC 822 MIME-like response. **All clients and all servers must obey this protocol.**

Computer Networks, 502 (emph. added). See Computer Networks, 502-507 (Section 7.3.4, “HTTP—The Hypertext Transfer Protocol”).

133. The Petitioner’s own reference (Fegan, Ex. 1005) describes HTTP as the standard protocol for client-server communications:

[0044] As used herein, the term “HTTP” refers to Hyper Text Transport Protocol that is the **standard World Wide Web client-Server protocol** used for the exchange of information (such as HTML documents, and client requests for such documents) between a browser and a Web server. HTTP includes a number of different types of messages that can be sent from the client to the server to request different types of server actions. For example, a “GET” message, which has the format GET, causes the server to return the document or file located at the specified URL.

Ex. 1005 (Fegan), [0044].

134. Accordingly, the fact that Gupta’s client computer 15 communicates with annotation server 10 using HTTP requests (e.g., commands) does not mean that annotation server 10 has a server-side *video editing environment* application performing cue point editing per Limitation 13.1. **The use of HTTP requests to communicate is simply a mechanism for the client to make a request (e.g., to provide content or to get content) and for the server to respond to the request (e.g., receive the content or provide the content).**

135. For example, a POSITA would understand that a **prior art** local cue point editing system, like that described in the Background of the Invention of the '286 Patent (see 2:16-25), where a client runs a specialized client-side application, would also use HTTP to communicate with servers receiving the edited video. Indeed, Gupta states in its background discussion that the client computers will use HTTP to communicate with annotation server **for all communications**. Gupta, [0032] ("The annotation server 10 communicates with the client computers 15 via ... the Hypertext Transfer Protocol (HTTP)"). This illustrates that the use of HTTP in Gupta does not disclose (or even suggest) that Gupta has a server-side *video editing environment* application performing cue point editing per Limitation 13.1.

136. Along the same lines, the Board preliminarily suggested: "In Gupta Figure 3, the Annotation Back End (151) 'translates user actions into commands destined for server 10,' and those commands are 'forwarded to annotation server 10 via HTTP requests' (HTTP Services 150) 'to perform the desired addition/retrieval.' Ex. 1004 ¶¶ 34, 44, 45." See Instn. Dec., 24-25. However, a POSITA would understand paragraphs [0044]-[0045] to refer to a typical client-web server architecture where Gupta's client 15 sends HTTP requests (*e.g.*, an HTTP "get," "post," or "put") to the annotation server 10, which provides an HTTP response. See Ex. 2023 (HTTP), 1-2; Ex. 2025 (Computer Networks), 13, 502-507; Fegan, [0044]. Indeed, Gupta states that computer 15 "communicates with media server 11 and

requests the identified media segment **using conventional HTTP requests.**” Gupta, [0099]. Therefore, a POSITA would readily recognize that the fact that ABE 151 sends HTTP requests/commands to annotation server 10 means that an edited annotation created by **client-side UIM 152** (e.g., using client side Edit Annotation of Figure 21) is transmitted using HTTP to annotation server 10 for storage in databases 17-18. This does not mean in any way that the ABE 151 is sending a command to initiate annotation editing operations to be performed by a *video editing environment* application at annotation server 10, as would be required by limitation 13.1.

137. A POSITA would understand that the client 15’s conventional use of HTTP does not change the fact that Gupta describes that the annotation creation and editing functionality is entirely client-side via UIM 152, which creates the locally-generated dialog boxes in Figures 7 and 21. Section VIII.D.1-D.4 explains in detail how the client-side UIM 152 module creates and edits annotations, and how Gupta does not disclose a server-side *video editing environment* application at annotation server 10.

138. The Petition misleadingly mixes quotes from paragraph [0034] with quotes from paragraphs [0044]-[0045] of Gupta. Paragraph [0034] states: “When a user of a client computer 15 desires to **add or retrieve annotations**, the client computer 15 contacts the annotation server 10 **to perform** the desired

addition/retrieval.” Gupta, [0034]. A POSITA would understand this passage as referring to the client making a request to **add** an annotation to databases 17-18 or **retrieve** an annotation from databases 17-18. Adding an annotation in this context means adding an annotation **created by client 15** (using the client-side application of Figure 7 for Add annotation), not creating by annotation server 10. This is consistent with [0032], which explains that the annotation server function is to store (add) and retrieve annotations. Gupta, [0032]. Read in context a POSITA would understand [0034] as indicating that the client computer 15 could (1) contact the annotation server to **add** to the backend databases 17-18 a new annotation locally created using the dialog box of Figure 7. As previously explained, the dialog boxes of Figure 7 (Add annotation) is generated by the client-side UIM 152 application, as described throughout this Section VIII.D. There is no server-side *VEE* application as claimed.

139. In summary, Gupta does not disclose Limitation 13.1 for a computer system ... comprising a video editing environment ... disposed at the server side of a public network.

6. Summary on Gupta’s Failure to Anticipate Limitation 13.1

140. I am informed that anticipation is an exacting standard. The Gupta reference within its four corners must explicitly or inherently disclose **each limitation of the claim**. *Net MoneyIn v. Verisign*, 545 F.3d 1359, 1371 (Fed. Cir.

2008). The Petitioner has not demonstrated that Gupta discloses the claimed server-side *video editing environment* application of Limitation 13.1. On the contrary, Gupta repeatedly discloses that its annotation editing functionality is provided by the client-side UIM 152 application, and thus does not meet Limitation 13.1.

E. Gupta Fails to Disclose *Modifying a Portion of the Cue Point Information at the Server Side ... According to the Cue Point Editing Command Received ... From the Client Terminal* (Limitation 13.5)

141. The Petitioner argues that Gupta anticipates Limitation 13.5 by disclosing “modifying a portion of the cue point information at the server side ... according to the cue point editing command received ... from the client terminal” based on Gupta’s disclosure of receiving a modified annotation at annotation server 10 **without saving the modified annotation**. See Pet., 25.

142. The Petitioner’s argument for Limitation 13.5 is based entirely on the incorrect premise that Limitation 13.5 provides for modifying **but not storing** the modified cue point:

The claim language does **not require saving the modified information or replacing the prior cue point information with the newly modified** information. EX1001, Claim 13; 19:26-35 (distinguishing “modifying” from “adding” and “deleting”). Therefore, a POSITA would have understood limitation [13.5] as requiring only modifying a portion of the existing cue points in accordance with the

received cue point editing command **and nothing further**. EX1002, ¶104.

See Pet., 25 n.5 (emph. added); Houh Decl., ¶ 104.

143. The Petitioner's and its expert's argument that "modifying" is distinguished from "adding" and "deleting," and thus "modifying" does not encompass storing the modified cue point information, is without merit. '286 Patent, 19:26-36. See Houh Decl., ¶ 104. "Modifying" is not defined to exclude storing, as the Petitioner suggests. A POSITA would readily understand that "modifying" encompasses storing the cue point in memory, such as a cue point stored in cue point server 114 or in the VEE 116 in Figure 1A of the '286 Patent.

144. The specification of the '286 Patent discloses that edited cue point information is stored, and may even be concurrently stored as editing occurs. This is consistent with the Patent Owner's application of the plain and ordinary meaning of the language of Limitation 13.5:

Further, cue point associations may be **updated** 326 in any useful way, including after submission, after exit from the VEE, or **substantially in real time**. In some embodiments, **the information may be maintained within files on a server. These files may be continually or periodically updated during the cue point editing process**. In other embodiments, the client may be working with very large files, large numbers of cue points, limited bandwidth, limited network reliability, or other technological limitations. In these environments, or for other reasons, it may be desirable or necessary to temporarily store

all information affected by the editor in local files. These local files may reside in one or more locations, including as temporary files on the client's terminal or in temporary files on the server computer.

'286 Patent, 16:24-37.

145. The claim language supports this construction because it provides for “*modifying a portion of the cue point information at the server side*,” which means that the modified cue point information must be stored at the server side computer system.

146. Additionally, Claim 13 provides for receiving a set of cue point information at the [server-side] video editing environment (13.2), modifying a portion of the cue point information at the server side [video editing environment] according to the cue point editing command (13.4-13.5), and then receiving a cue point selection ... identifying one of the cue points [from the modified cue point information] corresponding to a portion of the video information (13.6) resulting in streaming the portion of the video information to the client terminal ... in response to receiving the cue point selection (13.7).

147. The plain import of the claim language is that cue point information at the server-side *VEE* is received in limitation 13.3, and then it is remotely edited including being stored so that the edited cue point information can be selected to cause the corresponding video for the edited cue point information to be streamed to the client according to Limitations 13.6-13.7. If the edited cue point information was

not stored, as urged by the Petitioner, then the claimed method would not function as intended per its plain and ordinary language.

148. In my opinion, the Petitioner and its expert propose this unusual and incorrect claim construction that “*modifying ... the cue point information at the server side*” excludes storing the modified cue point information because Gupta has a deficiency, as explained further below.

149. Gupta does not meet Limitation 13.5 because it does not disclose that the edited annotation information in Figure 21 (“Edit annotation” 590 dialog box) is stored by annotation server 10 (part of the alleged *VEE*) as required by Limitation 13.5. See Gupta, [0144] (“FIG. 21 illustrates an exemplary ‘edit annotations’ dialog box 590. ... Dialog box 590 includes information and fields displaying the current information for the annotation”), Fig. 21. Gupta only discloses “**sending** the new annotation information to annotation server 10” (Gupta, [0100]), but it does not disclose storing the new annotation information (Gupta, [0144]) at annotation server 10.

150. The Petitioner’s expert cites Gupta’s paragraph [0048], which states that “Annotation content **authored** at client 15, using user interface 152, is received by ABE 132, and maintained in annotation content store 17.” Houh Decl., ¶ 105. However, paragraph [0048] does not reference the “Edit annotations” dialog box of Figure 21 upon which the Petitioner’s anticipation argument relies. Additionally,

paragraph [0048] refers to “annotation content **authored at client 15**” being “received by ABE 132 [at annotation server 10].” That does not disclose an annotation being modified at annotation server 10 in response to a “*cue point editing command*” received “*over the network*,” per limitation 13.5.

151. The Petitioner’s expert also cites paragraph [0034] which states: “When a user of computer 15 desires to add or retrieve annotations, the client computer 15 contacts the annotation server 10 to perform the desired annotation/retrieval.” Houh Decl., ¶ 106. This passage refers to storing or retrieving annotations from the backend memories 17-18, not to editing annotations which is the subject of limitation 13.5 (“*modifying a portion of the [received] cue point information*”).

152. A POSITA would view Gupta as not disclosing limitation 13.5, particularly that the edited annotation/cue point is stored. My understanding is that anticipation is an exacting standard and requires that the reference discloses all limitations exactly as arranged in the claim. Gupta does not meet this standard for limitation 13.5.

153. The Petitioner and its expert make no inherency argument that any of Gupta’s server-side components such as servers 10-13 and/or stores 17-18 would inherently store edited annotation information. As I have been instructed on the law, this means that Petitioner Aylo waives any inherency argument.

F. Gupta Fails to Disclose Limitation 13.2

154. Gupta does not disclose limitation 13.2 which recites the VEE “*receiving a set of video information*” at the VEE so that cue point information can be edited per limitations 13.4-13.5. A POSITA would understand Figure 1 and Figure 3 of Gupta, and the associated description as disclosing that client 15 receives video information, not that media server 11 (or any other component of the alleged VEE) is receiving video information. Gupta [0034], [0046] (“The user can select which one of different versions of multimedia content he or she wishes to receive from media server 11”).

155. Gupta discloses that streaming media server 11 (part of the Petitioner’s alleged VEE) “has access to streaming media content in the form of different media streams [that] might be stored as files in a database or other file storage system, while other media streams may be supplied to the server on a ‘live’ basis from other data source components through dedicated communication channels or through the Internet itself.” Gupta, [0030]. Gupta also states that “[w]hen a client computer 15 accesses a web page containing streaming media, a conventional web browser of client computer 15 also submits request to the media server 11 for streaming data.” Gupta, [0034].

156. Petitioner relies on both paragraphs [0030] and [0034] as allegedly disclosing limitation 13.2. However, both paragraphs describe that media server 11

can deliver streaming video received by **client 15**, not that streaming media is being received by **media server 11** (allegedly part of the *VEE*), as limitation 13.2 would require. Even in the case of a “live” media stream provided to media server 11, there is no disclosure that the media stream is being received so that its cue point information (e.g., annotations) can be edited, as limitations 13.2 and 13.4-13.5 would require.

157. Petitioner and its expert reference paragraph [0034] of Gupta, but the client 15 receiving video does not equate to the media server receiving video, which is what limitation 13.2 would require. For example, the media server could be delivering video to the client device based on video at the media server. Thus, the media server would not be receiving video.

158. Petitioner and its expert also refer to Figures 22-23 of Gupta, which depict a GUI “that concurrently displays annotations and corresponding multimedia content” at client 15. Gupta, [0145]-[0155]. See, e.g., Gupta, [0152] (“[a] user can browse through the visual content of the media content”). Figures 22-23 thus reflect the client device receiving video information, not multimedia server 11 (or any other part of the alleged *VEE*) receiving video information, as limitation 13.2 requires.

G. Gupta Fails to Disclose Claim 15 For *Receiving an Edit Mode Entry Signal ... From the Client Terminal Representing Entry into the Cue Point Edit Mode, and Switching the Video Editing Environment to the Cue Point Editing Mode in Response to Receiving the Edit Mode Entry Signal*

159. Claim 15 recites an *edit mode entry signal* sent over the network from the client to the server-side that is separate from the *cue point editing command* of Limitation 13.4, which is also sent from the client over the network. The VEE “switch[es] ... to the cue point editing mode in response to receiving the edit mode entry signal.” Thus, Claim 15 calls for two separate signals sent from the client to the server-side application: (1) an *edit mode entry signal* and (2) the *cue point editing command*.

160. A POSITA would understand from the claim language that the *edit mode entry signal* is a signal received from the client over the network that causes the *video editing environment* to switch to the *cue point editing mode*. The *edit mode entry signal* that triggers entry into the *cue point editing mode* is disclosed, for example, as a pause signal or another client-side signal in the '286 Patent specification. For example, the specification states:

In some embodiments, the method further provides a **cue point editing mode** configured to perform at least one of creating, modifying, deleting, or interpreting at least a portion of the set of cue point information. In certain of these embodiments, the method further **receives a pause signal over the network** representing that playback of the client video data has been paused, and transmits [into] the cue point editing mode **after receiving the pause signal**. In certain others of these embodiments, the method further **receives a pause signal over the network** representing that playback of the client video data has

been paused, receives a time position representing the time position within the client video data at which playback was paused, and creates a cue point location associated at least in part with the client video data and the time position.

'286 Patent, 3:48-61 (emph. added). See '286 Patent, 15:26-39 (Fig. 3A: 320 “Poll for Cue Point Mode” to enter cue point edit mode), 18:18-41 (Fig. 3B: 254 “Poll for Interactivity” to enter cue point edit mode). See also Ex. 1010 ('286 File History), 329 (original Cl. 16: “receiving an edit mode entry signal over the network representing entry into the cue point edit mode”).

161. Therefore, consistent with Claim 15, the specification discloses transitioning into the *cue point editing mode* in response to the server-side receiving an *edit mode entry signal* transmitted from the client over the network. There is no such disclosure in Gupta.

162. The Petitioner argues that the claimed *edit mode entry signal* is the user selecting the “edit” option in selection box 446 of the “View annotations” dialog box of Figure 15 of Gupta. Pet., 30 (citing [0127]-[0127] of Gupta); Houh Decl., ¶¶ 120-128. However, selecting “edit” is a local input to a local application (UIM 152) running on client 15; it is not the transmission of a signal by client 15 *over the network* to the server-side alleged *VEE* (allegedly annotation server 10, servers 11-13, databases 17-18), as required by Claim 15.

163. Specifically, a POSITA would understand that selecting the “edit” option in Figure 15 causes the locally-generated View annotations dialog box of Figure 15 to transition to the locally-generated “Edit annotations” dialog box of Figure 21. As I discussed previously, both Figure 15 and Figure 21 are locally-generated by the UIM 152 software executed on client 15. See Gupta, [0045], [0081] (local client UIM 152 provides “[n]umerous windows, dialog boxes, toolbars, etc. [to] be displayed to the user by interface module 152.”) [0082] (“The interface module 152 is an application program ... executed at client 15”). See Section VIII.D.2 for a detailed explanation on how the client-side UIM 152 program locally generates the dialog windows of Figures 15 and 21.

164. A POSITA would thus readily understand that the selection of “edit” in Figure 15 is a **local input** to the local UIM 152 application, causing it to generate the new Edit annotations dialog box of Figure 21. See Gupta, [0128] (“Selection of the ‘edit’ feature [in Fig. 15] causes interface module 152 to display another dialog box [Fig. 21] allowing the user to edit the content of the annotation, as discussed in more detail below”), [0144] (“FIG. 21 illustrates an exemplary ‘edit annotation’ dialog box 590. Edit annotation dialog box 590 is displayed in response to a user request to edit an annotation, such as selection of the ‘edit’ option in menu 466 of FIG. 15.”), Fig. 15, Fig. 21. A POSITA would thus understand that selecting “edit” in the Fig. 15 dialog box of Gupta does not cause the transmission of a signal from

the client 15 over the network to the alleged server-side *VEE* (servers 10-13, stores 17-18). The local input of the user clicking a mouse to select “edit” is not a signal transmitted from the client over the network, and thus does not meet Claim 15.

165. For these reasons, I disagree with Dr. Houh’s assertion in ¶¶ 120-128 of his declaration suggesting that selecting “edit” causes a client terminal to send a signal (*edit mode entry signal*) over the network to the server-side (alleged *VEE*: servers 10-13, stores 17-18). Specifically, I disagree with Dr. Houh’s assertion (in ¶ 122 of his declaration) that a user selecting “edit” in Figure 15 sends a signal from the client “over the network” to “switch[] the software executed by servers 10-13 and stores 17-18 to display the ‘Edit Annotation’ dialog box,” as he puts it. Houh Decl., ¶ 122. Additionally, Gupta does not disclose that annotation server 10 responds to the (non-existent) *edit mode entry signal* by “provid[ing] HTML pages to display the ‘Edit Annotation’ dialog box on the client computer,” as asserted. Houh Decl., ¶125. I disagree because, as explained above, both the View Annotations dialog box of Figure 15 with the “edit” option and the Edit Annotations dialog box of Figure 21 are client-side dialog boxes generated by the client-side UIM 152 software. They are not generated by server-side software as Dr. Houh asserts. See Section VIII.D.1-D.4.

166. The Petitioner and its expert also cite paragraphs [0034], [0044] and [0047] of Gupta regarding Claim 15. Pet., 30-31 (citing Houh Decl., ¶¶ 120-128).

167. Paragraph [0034] generally describes that client 15 has a web browser that can contact a web server 12 to request an HTML page, and that it can also submit requests to media server 11 to receive streaming media. See Pet., 31. Paragraph [0034] does not address the alleged *cue point editing* mode allegedly found in Figure 21 of Gupta and allegedly triggered by selecting “edit” in Figure 15.

168. Paragraph [0044] of Gupta describes that, as with any client-server interaction over the Internet, the client 15 communicates with servers by formulating “commands” as HTTP requests. See Pet., 31; Houh Decl., ¶ 124. It does not address the alleged *cue point editing mode* allegedly found in Figure 21 of Gupta and triggered by selecting “edit” in Figure 15. The fact that client 15 communicates with servers using HTTP requests does not remotely establish that the “edit” selection in Figure 15 causes a signal to be sent *over the network* to the alleged server-side *VEE*. See Section VIII.D.5 (addressing the standard use of the HTTP request-response protocol in client-server communications).

169. Paragraph [0047] of Gupta indicates that annotation server 10 has a server-side HTTP services module 131 to manage client-server HTTP communications with client 15. It does not discuss, let alone disclose, the “edit” selection in the Figure 15 dialog box causing the generation of the Figure 21 dialog box, let alone does it disclose that selecting “edit” causes the transmission from the client *over the network* of a *edit mode entry signal* to the alleged server-side *VEE*.

170. I also disagree with this assertion by Dr. Houh:

Therefore, in Gupta, when a user selects the “edit” option for a selected annotation, **annotation server 10 receives an edit mode entry signal** to enter a cue point edit mode and **provide HTML pages** to display the “Edit Annotation” dialog box on the client computer 15’s web browser.

Houh Decl., ¶ 125. As I explained in detail in Section VIII.D.1-5, the selection of “edit” in View annotations dialog box of Figure 15 causes the client-side UIM 152 to generate the Edit annotation dialog box of Figure 21. The selection of “edit” is a local input to the client-side UIM 152 “application program,” not a signal sent over the network to the annotation server 10. I note that Dr. Houh’s assertion that “annotation server 10 receives an edit mode entry signal” is not supported by citation to Gupta. As I understand matters, that means it should be given little to no weight, particularly in view of the significant contrary evidence I have provided from Gupta. Further, nowhere does Gupta disclose or remotely suggest that the annotation server responds to the (non-existent) edit signal by “provid[ing] HTML pages to display the ‘Edit Annotation’ dialog box on the client computer.” See Houh Decl., ¶ 125. Again, that statement is not supported by citation to evidence, and it is contradicted by the evidence I have cited previously.

171. Accordingly, Claim 15 is not anticipated by Gupta.

H. Gupta Fails to Disclose Claim 18 For *Receiving, Over the Network, Authentication Information Related to Access Authority of the Remote Client Terminal*

172. For anticipation, Gupta must disclose, either expressly or through inherency, that the *video editing environment* receives “*over the network, authentication information related to access authority*” of the client terminal. Gupta does not disclose that feature.

173. The Petition cites to the following passages of Gupta for Claim 18:

[0007] Annotations can be added by anyone with appropriate access rights to the annotation system.

[0090] According to one embodiment of the invention, annotation server 10 of FIG. 3 supports read and write access controls, allowing the creator of the set to identify which users are able to read and/or write to the annotation set.

[0046] Client 15 also executes a web browser module 153, which provides a conventional web browsing interface and capabilities for the user to access various servers via network 16 of FIG. 1. Web browser 153 also provides the interface for a user to select particular media streams for presentation. The user can select which one of different versions of multimedia content he or she wishes to receive from media server 11 of FIG. 1. This selection can be direct (e.g., entry of a particular URL or selection of a “low resolution” option), or indirect (e.g., entry of a particular desired playback duration or an indication of system capabilities, such as “slow system” or “fast system”). Alternatively, other media presentation interfaces could be used.

[0052] Author field 182 [in Figure 4 depicting metadata for an annotation] contains data identifying the user who created annotation entry 180 and who is therefore the author of the annotation. The author is identified by ABE 151 of FIG. 3 based on the user logged into client 15 at the time the annotation is created.

[0109] According to one embodiment of the invention, annotation server 10 of FIG. 3 supports read and write security, allowing the creator of the set to identify which users are able to read and/or write to the annotation set. In this embodiment, only those sets for which the user has read access are displayed in set list 376. See Gupta, Fig. 12 (Query annotations dialog box).

Pet., 31.

174. Collectively, these passages disclose that the creator of an annotation can create read/write access rights for the annotation. They do not expressly disclose that read/write access rights (or associated credentials) are transmitted over the network to the annotation server 10. The assertion in this ground is anticipation, not obviousness, and my understanding is that anticipation requires the literal disclosure of the limitation, not a showing that the limitation is purportedly obvious.

175. The Petitioner has not argued inherency. Even for the sake of argument, it is not inherent that Gupta discloses that that *authentication information* is transmitted over the network from the client. For example, a POSITA would understand that authenticated users with read/write access rights could be preauthorized at the annotation server such that there is no need for the

authentication information to be transmitted over the network to the annotation server. Alternatively, *authentication information* could be directly received into annotation server 10 by a service representative directly entering the information from a local terminal. Accordingly, it is not inherent that Gupta discloses receiving *authentication information* from the client.

176. The Petitioner also cites to paragraphs 129-133 of the Houh Decl. Pet., 31. Paragraph 132 of the Houh declaration argues that “the computer system includes some form of authentication information,” such as an IP address or username/password. However, no evidence is provided by Dr. Houh in support of this alleged “common practice.” Also, there is no showing that the alleged *authentication information* (IP address or username/password) is disclosed in Gupta as being transmitted over the network to enforce read/write access rights. In fact, Gupta discloses nothing regarding authentication using IP addresses or user names/passwords.

177. Accordingly, Claim 18 is not anticipated by Gupta.

I. Claims 14 and 17-20 Are Patentable Over Gupta

178. Claims 14 and 17-20 are patentable over Gupta for at least the same reasons set forth for Claim 13.

IX. CLAIMS 13-15 AND 18-20 ARE NOT ANTICIPATED BY FEGAN (GROUND 2)

A. Claims 13-15 and 18-20 Are Patentable Over Fegan

179. Fegan fails to anticipate the following limitations of Claim 13 of the '286 Patent:

- “*providing a computer system ... comprising a video editing environment, and being disposed at a server side of a public network*” (Limitation 13.1); and
- “*receiving a cue point editing command over the network from a remote client terminal*” and “*modifying a portion of the cue point information at the server side ... according to the cue point editing command*” (Limitations 13.4-13.5).

B. Fegan Fails to Disclose A Computer System ... Comprising a Video Editing Environment, and Being Disposed at a Server Side of a Public Network Per Limitation 13.1

1. Fegan Does Not Disclose a Server-Side Video Editing Environment Application Per Limitation 13.1

180. As discussed above in Section VIII.C, the inventor repeatedly explained during the prosecution of the '286 Patent the *video editing environment* as a being server-side application(s) for performing editing of cue point information that enables **remote cue point editing** by the client terminal. This understanding is supported by the '286 Patent specification and the language of Claim 13, also discussed in Section VIII.C.

181. I note that in its Institution Decision for the -00710 IPR, the Board raised a question as to whether Fegan discloses Limitation 13.1. See Ex. 2034

(Decision on Institution, Paper 13, 9/5/2024), 33 (calling into question “whether Fegan discloses Limitations 13.1, 13.4, 13.5, and Claim 15”). The Board asked “whether the segmenting and annotation tools of the browser-based media player are executed on the client side or server side of the network.” Ex. 2034, 33. As explained below, the segmenting and annotation tools in Fegan are applications run on the client, not the server. An examination of Fegan and Petitioner’s arguments demonstrates that Fegan does not disclose Limitation 13.1.

182. Similar to Gupta, Fegan discloses a system that allows a user to create “segments” (bookmarks) by defining start/end points in a selected multimedia file. The segments can include annotations such as comments. See Fegan, Figs. 3-4 (an audio file is selected, and a segment named “flint sit down strike” is defined and annotated). The Petitioner points to such segments/annotations as disclosing cue point information and further points to the utilities for editing them as the alleged “*video editing environment ... disposed at a server side of a public network.*” Fegan does not support that assertion.

183. I note that Fegan makes some vague high-level introductory statements that preferably “[t]he technical tools for **carrying out the multimedia presentation** are maintained on the server-side.” Gupta, [0007] (cited by Pet, 35). Fegan also states that “[i]n preferred embodiments, the systems and methods are hosted in an entirely server-side environment.” Fegan, [0008] (cited by Pet., 35). However, after

maintaining these background statements, Fegan proceeds to disclose **saving** segments/annotations at remote servers and allowing clients to **retrieve** segments/annotations from a remote server. Yet, when it comes to **editing** annotations, all of Fegan’s disclosed embodiments carry out annotation editing operations locally through **only client-side software**. Figures 5-7 of Fegan, discussed in detail below, are the windows enabling the **editing** of annotations, and they are implemented entirely through client-side software (e.g., a client browser and a client media player). See, e.g., Fegan, [0074] (client-side “browser load[s] the file into a browser-based multimedia player”), [0076] (“browser window ... and a compatible multimedia player are displayed”), [0080] (“browser window for editing of multimedia content that includes a multimedia file loaded into a compatible multimedia player”). Fegan does not disclose a server-side application for performing editing of annotations based on commands sent by a client device. The editing of annotations in Fegan is not remote cue point editing using a *video editing environment* application at the *server side of a public network* per limitations 13.1-13.5.

2. Fegan’s Reference to MYSQL Does Not Disclose a Server-Side Video Editing Environment Application Per Limitation 13.1

184. The Petitioner asserts that there may be server-side database software in Fegan such as MYSQL or PHP. Pet., 34 (citing Fegan, [0010]); Ex. 1002 (Houh

Decl.), ¶ 154. Fegan does not disclose that MYSQL or PHP programming is executed server-side to perform annotation editing per limitation 13.1. Further, as described in Fegan, and as would be well understood by a POSITA, database software like MYSQL is used to maintain information in a database, i.e., it provides for the **storage and retrieval** of information. Ex. 2012 (What is MYSQL?), 1-2; Ex. 2030 (Willams), 8:22-23 (“MySQL stores and retrieves information using standard SQL commands.”); Ex. 2029 (Hensen), [0066] (“A MySQL database server is provided at the datastore [] to facilitate ... storage and retrieval” of data). Fegan confirms this understanding because it discloses that such database software is used to “manage[] the appropriate database.” Fegan, [0010]. A POSITA would thus understand that this database software is performing server-side **storage and retrieval** operations, not server-side **editing of annotations** as claimed. There is no disclosure by Fegan that such database software performs any server-side **editing of annotations** based on commands sent from a client terminal. Fegan, [0010]. In summary, Fegan does not disclose that MYSQL or other database software provides a server-side application for editing annotations based on client commands, as required by the server-side *video editing environment* application of limitation 13.1.

185. Similarly, Fegan’s reference to the general use of PHP and Javascript (Pet., 34) at the server does not disclose a server-side “*video editing environment*” application for the server-side editing of annotations based on client commands, as

would be required by limitation 13.1 (and limitations 13.4-13.5). Fegan, [0010]. See Pet., 34; Houh Decl., ¶¶ 154-155.

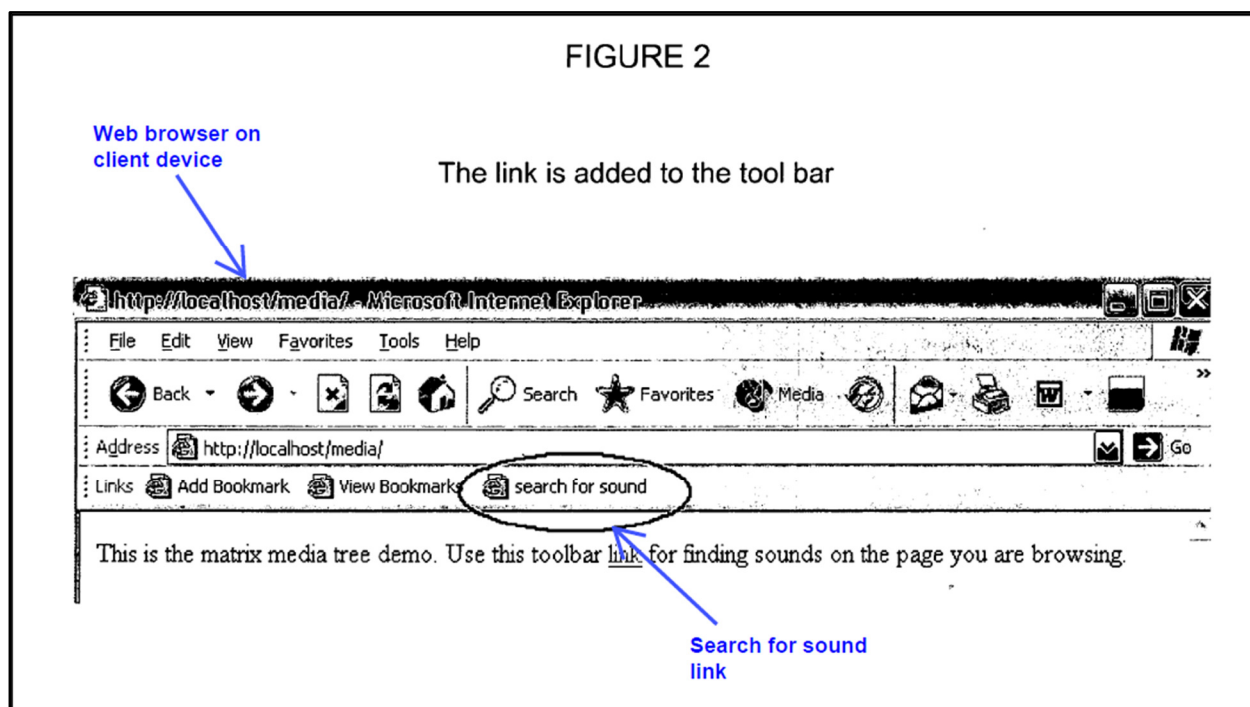
3. Fegan's References to Programs Stored on Remote Servers for the Search Window of Figure 2 Does Not Disclose a Server-Side *Video Editing Environment* Application Per Limitation 13.1

186. The Petitioner and its expert also argue that the server-side *video editing environment* application of Limitation 13.1 is disclosed by Fegan in various passages referencing (1) programs stored on remote servers, or (2) the storage of segment and annotation information on remote servers. Pet., 34-35 (citing Fegan, [0071], [0073]-[0074], [0083], [0007], [0009], and Figs. 1-7).² Neither Fegan's description of programs stored on remote servers or its description of storing segment/annotation information on remote servers discloses Limitation 13.1 for a server-side *video editing environment*.

187. Regarding paragraph [0071], it pertains to Figure 2 of Fegan depicting a multimedia search window. Fegan, Fig. 2, [0071]. See also Fegan, Fig. 1. Fegan describes that the client web browser can be used to search for and locate content on

² Each of the passages and figures cited by the Petitioner are addressed in my opinions. See Pet., 34-35 (citing Fegan, [0071], [0073]-[0074], [0007], [0009], Figs. 1-7).

the Internet, such as audio files. Fegan, [0071]. Fegan describes initiating a search by selecting a “search for sound” hyperlink placed on the web browser which is currently connected to the web site at //localhost/media/:



Fegan, Fig. 2 (annotated in blue), [0070]-[0071].

188. In that case, the “search for sound” hyperlink on the browser is selected to access a program “stored in one or more remote locations” to be used in connection with conducting the search. Fegan describes the program as allowing the **client browser functionality** to be “extended” or “added to” Fegan, [0071] (“the functionality of the browser is extended;” “functionality for locating multimedia content is added to a browser”). Fegan goes on to explain that the hyperlink connects the user to a remote server that provides “one or more applets” used to “search[] for

multimedia content.” Fegan, [0071]. A POSITA would understand that as meaning that the applets are downloaded from the remote server to the client and executed by the browser at the client. It was well known in this timeframe that applets were essentially mini-programs that could be stored at a server to be downloaded and executed by the client device by or through the client browser. See Ex. 2025 (Computer Networks), 501-502, 631-632; Ex. 2027 (U.S. Pat. No. 7,216,300, Dang, to NCR), 1:24-25 (“Applets are programs executed as part of a Web page and displayed within a Java-enabled browser”); Ex. 2026 (U.S. Pat. No. 6,094,677, Capek, to IBM), 2:38-41 (“Programs written using Java can be downloaded over the WWW in the form of byte codes for execution on a Java virtual machine at the client 14. Such programs are known as ‘applets’”); Ex. 2028 (Java Applet – Wikipedia), 1 (“The code of the applet is downloaded from a web server and the browser either embeds the applet into a web page or opens a new window showing the applet’s user interface. ... Java applets can be executed by browsers for many platforms, including Windows, Unix, Mac OS and Linux”).

189. Accordingly, paragraph [0071] and Figure 2 disclose a browser window generated by a **client-side browser application** extended with a client-side applet program used to **search for multimedia content** on the Internet such as audio files. The Petitioner’s and its expert’s argument based on paragraph [0071] for limitation 13.1 thus fails for at least two reasons. See Pet., 34; Houh Decl., ¶ 154. First, Fegan’s

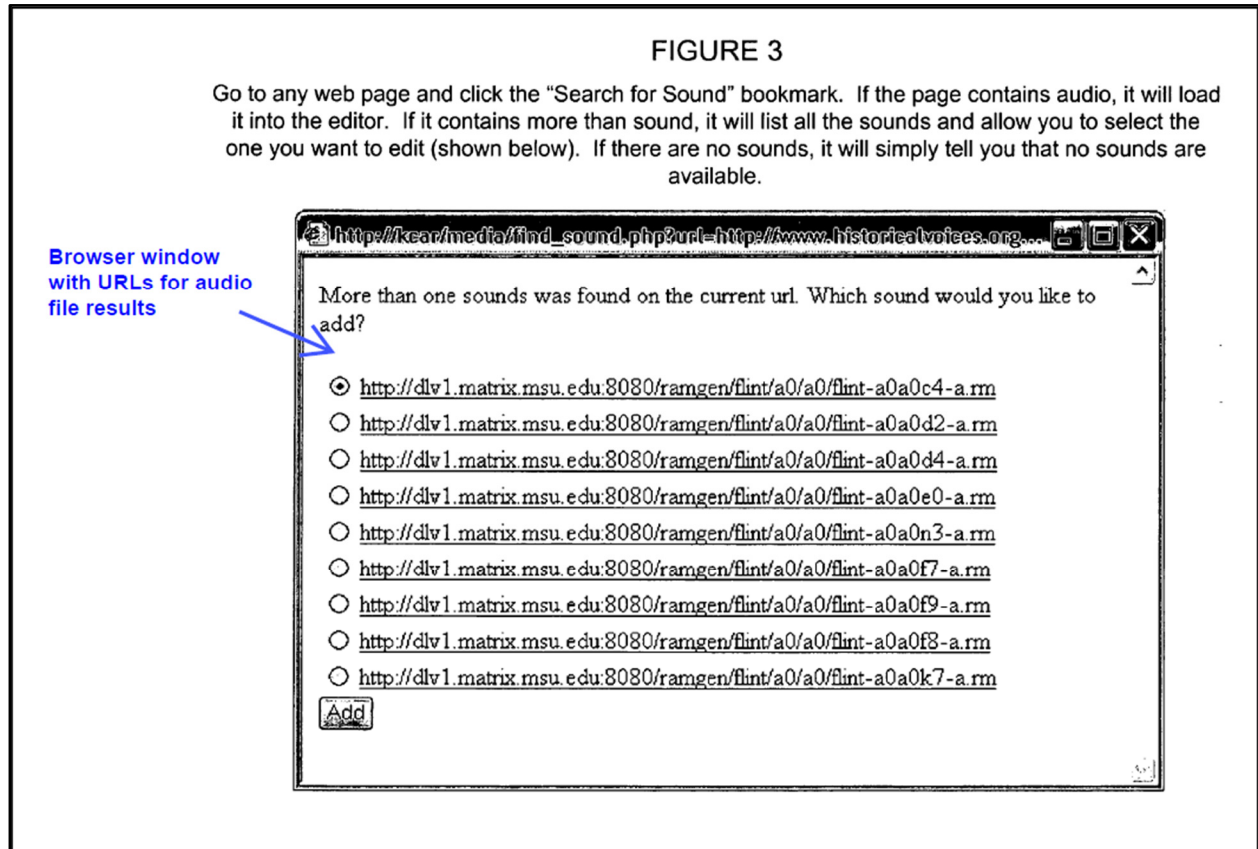
Figure 2 and paragraph [0071] do not relate to a **server-side video editing environment** providing for remote cue point editing because Fegan is describing only the execution of **client-side applications**. Second, the browser of Figure 2 including any program from a remote location is used for **searching** for multimedia content (e.g, an audio file as depicted in Figure 3), **not for editing annotations** as would be required by limitation 13.1. Accordingly, the browser search window of Figure 2 and paragraph [0071] do not disclose the client-side *video editing environment* of limitation 13.1.

4. Fegan’s Reference to Programs Stored on Remote Servers in Connection with the Search Results Window of Figure 3 Does Not Disclose a Server-Side Video Editing Environment Application Per Limitation 13.1

190. The Petitioner and its expert allege that the “software on a remote server” referenced in paragraph [0073] for Figure 3 of Fegan discloses limitation 13.1. Pet., 34; Houh Decl., ¶ 155. As explained below, the referenced “software on a remote server” pertains to **saving search results** (e.g., audio files or other multimedia files from the search using Figure 2), **not performing editing of annotations** as per limitation 13.1. The operations conducted for Figure 3 and paragraph [0073] are described below.

191. After a user selects the “search for sound” hyperlink on the client browser window of Figure 2, Fegan discloses the client browser generating another

browser window as depicted in Figure 3, which is a search results window providing a list of URL links to audio files:



Fegan, Fig. 3 (annotated).

192. Figure 3 discloses that the search results (audio files) are URLs identifying the location of each found audio file:

[0073] After a location on a computer network is searched, the **results of the search** are displayed in a window [Figure 3]. Any multimedia content found at the searched location can be displayed. The display can be arranged by media type (e.g., video, audio, image, text, etc.), file type, file name, location (e.g., URL), or by any other desired means. FIG. 3 shows a **browser window in which the results of a search for**

multimedia content on a web page are displayed by location. In preferred embodiments of the present invention, **the data comprising the search results may be stored by software on a remote server.**

The stored data includes information useful to locate, identify, and categorize the multimedia files, but preferably excludes the multimedia files themselves. As shown in FIG. 3, a means is provided for a user to select one or more of the identified multimedia files for access.

Fegan, [0073] (emph. added), Fig. 3.

193. Accordingly, a POSITA would understand the “software on a remote server” as being used to **save search results**, not to perform editing of annotations as would be required by limitation 13.1. The argument set forth by the Petitioner and its expert regarding “software on a remote server” does not pertain to editing of annotations. See Pet., 34; Houh Decl., ¶ 155. Accordingly, the search results window of Figure 3 including paragraph [0073] do not disclose a server-side *video editing environment* application for editing cue points (annotations) per Limitation 13.1.

5. Fegan’s Reference to “Software Stored on a Remote Server” to Save Segments and Annotations Does Not Disclose a Server-Side *Video Editing Environment* Application Per Limitation 13.1

194. The Petitioner and its expert assert that the reference in paragraph [0083] to “software stored on a remote server” to save annotation information meets Limitation 13.1. Pet., 34; Houh Decl., ¶ 155. However, this passage is directed to

storing annotation information, not **editing annotations** as required by limitation 13.1.

195. Paragraph [0083] is entitled “Organizing Multimedia Content as a Presentation” and describes how the multimedia files, segments, and annotations are organized:

[0083] The systems and methods of the present invention may be used to create user-customizable **multimedia presentations comprising multimedia files, segment information, and annotative information**. In preferred embodiments, software that is stored on a remote server allows a user to save and store segment and annotative information associated with one or more multimedia files. Because the systems and methods of the present invention utilize time and/or spatial-based index points and annotations, multimedia presentations may be created without the need to permanently store multimedia files, or divide them into separate files corresponding to each selected “clip.” In some embodiments, software that is stored on a remote server provides one or more templates upon which a multimedia presentation may be based. In preferred embodiments, both preset and customizable templates are provided. In other embodiments, the multimedia presentation is executed by means of a specialized programming language. In particularly preferred embodiments, the multimedia presentation is executed by means of Synchronized Multimedia Integration Language (“SMIL”). In yet other embodiments, the multimedia presentation is executed by means of Hypertext Markup

Language (“HTML”), although any suitable programming language may be used.

Fegan, [0083] (emph. added).

196. Accordingly, paragraph [0083] describes that “software stored on a remote server” is used for **storing** annotations, **not editing annotations** as incorrectly asserted by the Petitioner and its expert. See Pet., 34; Houh Decl., ¶ 155. Thus, this portion of Fegan does not disclose a server-side *video editing environment* application for editing cue points (annotation) per limitation 13.1,

6. Fegan’s Segment/Annotation Creation Window of Figure 4 Based on the Client-Side Multimedia Player and the Client-Side Browser Does Not Disclose a Server-Side *Video Editing Environment* Application Per Limitation 13.1

197. The Board asked whether the segmenting and annotation tools of Fegan are executed client side or server side. Instn. Dec., 33. The Board noted, correctly, that “Fegan discloses the use of a browser-based multimedia player application that has editor controls available for use on the client side to annotate and edit video files.” Instn. Dec., 31. The Board also correctly noted that Fegan discloses that the tools for segmenting video files and creating annotations are contained in the client-side browser. Instn. Dec., 31 (citing Fegan, [0074], [0076], [0078], [0080]). As explained further below, Fegan discloses that the tools for creating segments and annotations run on the client side, not the server side.

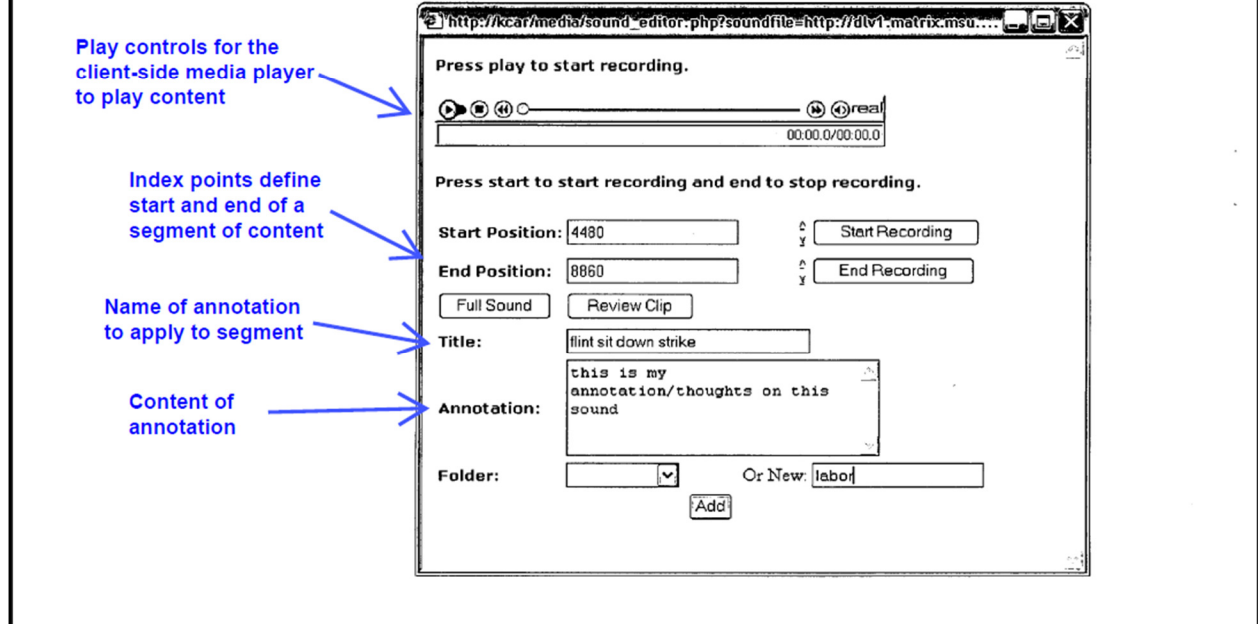
198. Fegan discloses that a user can select an audio file in the search results browser window of Figure 3, which causes the selected audio file to be loaded into a **browser-based media player running on the client** as depicted in Figure 4. Fegan, [0074]-[0076], Fig. 4. A POSITA would understand Figure 4 to depict a segment/annotation creation window:

[0074] After a desired multimedia file has been selected [from Figure 3], software directs **the browser to load the file** into a browser-based **multimedia player** that is compatible with the file type of the selected multimedia file. Examples of browser-based multimedia players that may be used with the present invention include RealPlayer (Real Networks), Windows Media Player (Microsoft), and QuickTime Player (Apple), although any browser-based multimedia player may be used.

Fegan, [0074] (emph. added). Figure 4 is depicted below:

FIGURE 4

The system will load the sound into the editor and allow you to create a segment by pressing start and end recording and then adding a title, annotation, and dictating what folder this annotation should go in on your portal page.



Fegan, Fig. 4 (annotated in blue).

199. Fegan states that Figure 4 is a “browser window in which a user-selected multimedia file and compatible multimedia player are displayed.” Fegan, [0076]. The top of Figure 4 states that “the system will load the sound into the editor and allow you to **create a segment** by pressing start and end recording and then **adding a title, annotation**, and dictating what folder this annotation should go in your portal page.” Fegan, Fig. 4. These and other statements in Fegan confirm that Figure 4 including the “editor” is a client-side user interface that is generated by a **client-side software** . Fegan, [0074] (“load the file into a **browser-based multimedia player**”).

200. A POSITA would accordingly understand that the “editor” of Figure 4 is **client-side** software for **creating** segments/annotations, not a server-side *video editing environment* for editing cue points. First, the software is disclosed as being client-side, not server-side, as required by limitation 13.1. Second, the software provides for creating annotations, not editing annotations as would be required by limitation 13.1.

a. A POSITA Would Understand That Fegan’s “Browser-Based Multimedia Player” is Client Software

201. A POSITA would readily understand from Fegan’s reference to a “browser-based multimedia player” that the multimedia player is client-side software executed by the client. Fegan, [0074] (“load[s] the file into a browser-based multimedia player”).

b. Multimedia Players Were Understood in the Art as Client Applications

202. It was well understood in the art that multimedia players such as those mentioned by Fegan (Fegan, [0074]: Windows Media Player, RealPlayer, etc.) were client applications executed by the client. Ex. 2031 U.S. Pub. No. 2003-0182139A1 (Harris, to Microsoft), Abstract (“the client computer executes a media player”), [0027] (“Client computer 152 also executes a web browser 164 and a media player application program 166”); Ex. 2033 U.S. Pat. No. 7,797,242 (Gautier, to Apple), 3:23-25 (“each of the client media player programs being resident on a different

client machine”); Ex. 2032 (Krikorian, to Sling Media), Abstract (“media player running on any of a wide range of client devices for viewing the media”).

c. Figure 4 Reflects Applications for Creating Annotations, Not Editing Annotations, And the Applications Are Client-Side, Not Server-Side

203. Fegan discloses that segments and associated annotations can be created using the client-side browser and media player applications generating the window of Figure 4 as follows (all emph. added):

[0076] The selected multimedia file and **compatible multimedia player are then displayed in a browser window**. FIG. 4 shows a browser window in which a user-selected multimedia file and a compatible multimedia player are displayed. The window is configured to provide user access to the playback controls of the multimedia player. In preferred embodiments, the playback controls include the functions typical to most multimedia players, including, but not limited to, the ability to start, stop, and pause playback in real time, move in forward and reverse at speeds faster and slower than real time (e.g., fast-forward, rewind, slow motion, etc.), and adjust playback conditions (e.g., audio volume, image brightness and contrast, etc.).

[0077] D. Segmenting Multimedia Content

[0078] As shown in FIG. 4, a **user-selected multimedia file and compatible multimedia player** are directed by software to be **displayed in a browser window**. In preferred embodiments, the browser window contains **additional tools for segmenting multimedia files**. In particularly preferred embodiments, the

segmenting tools allow a user to generate time-based index points for a multimedia file (e.g., spatial parameters for image files). The use of time-based-spatial based index points provides a means for the virtual segmentation of a multimedia file, allowing a user to create “clips” of specified portions. Playback of the multimedia file may be started or stopped at any point in the file’s timeline. The data associated with the time-based index points may be **saved and stored on a remote server, and may be retrieved from the remote server and used whenever the associated multimedia file is accessed.** Because the systems and methods of the present invention utilize time-based/spatial-based index points, multimedia editing may be achieved without the need to permanently store multimedia files, or divide them into separate files corresponding to each selected “clip.” The only data required to be stored for the purpose of segmenting multimedia files is the data associated with the time-based index points.

[0079] E. Annotating Multimedia Content

[0080] The systems and methods of the present invention provide a means for a user to **add annotative information** to a multimedia file. In preferred embodiments, such user created **annotative information includes a title, annotative notes, and a designated location for storage and retrieval of the annotative information.** FIG. 4 shows a browser window for editing of multimedia content that includes a **multimedia file loaded into a compatible multimedia player,** controls for generating time-based index points for virtual segmentation of the multimedia file, and a means of **entering annotative information.** In preferred embodiments, such **annotative information**

is entered via one or more text fields in a browser window.

Annotative information typically comprises text information, although any type of multimedia data may be used (e.g., audio, video, image, etc.). User-generated annotations may be used for any annotative purpose, including, but not limited to, providing contextual information, critical commentary, or descriptive information about the associated multimedia file.

Fegan, [0076]-[0080] (emph. added).

204. Accordingly, Fegan discloses in connection with Figure 4 the use of **client-side applications** (browser and media player applications) with client-side tools to create segments of media and to create associated annotations. A POSITA would understand that the segmenting and annotation tools are executed client-side. There is no disclosure (or even a suggestion) that those tools are executed server-side in Fegan. This answers the Board's question about whether the segmenting and annotation tools are executed on the client side or server side of the network. See Instn. Dec., 33.

205. Figure 4 does not disclose limitation 13.1 because it discloses the use of **client-side applications** (browser and media player) for the local creation of annotations, not a **server-side** *video editing environment* application.

206. Fegan discloses that after a segment/annotation has been locally created at the client using the client-side browser and media player applications in Figure 4, the segment and an associated annotation can be uploaded to a remote server. Fegan

states: “The data associated with the time-based index points may be **saved** and stored on a remote server, and may be **retrieved** from the remote server and used whenever the associated multimedia file is accessed.” Fegan, [0078]. See Fegan, [0080] (“a designated location for **storage and retrieval** of the annotative information”), [0083] (software at remote server allows a user to “**save and store** segments and annotative information”). This is further confirmation that the segmenting/annotation operations do not occur server-side. This is because Fegan discloses that a remote server can be used for the **storage and retrieval** of annotations (cue points), but **not for creating annotations**. The Petitioner’s and its expert’s various arguments citing these passages relating to storage of annotations on remote servers are off-point. See, e.g., Pet., 34; Houh Decl., ¶ 155. Nowhere does Fegan disclose a **server-side video editing environment** application for performing the remote editing of an annotation by a client as called for by limitation 13.1.

7. Fegan’s Discussion of the Portal Window of Figure 5 Does Not Disclose a Server-Side Video Editing Environment Application Per Limitation 13.1

207. The Petitioner asserts that the “portal” on a remote server in Fegan meets Limitation 13.1. Pet., 35 (citing [0009] (“each user has a personal ‘portal’ hosted on the system”)); Houh Decl., ¶ 155. Figure 5 of Fegan shows the portal of paragraph [0009]. Fegan describes that after the segment/annotation is created according to Figure 4, it can be stored at a folder such as in the user’s portal depicted

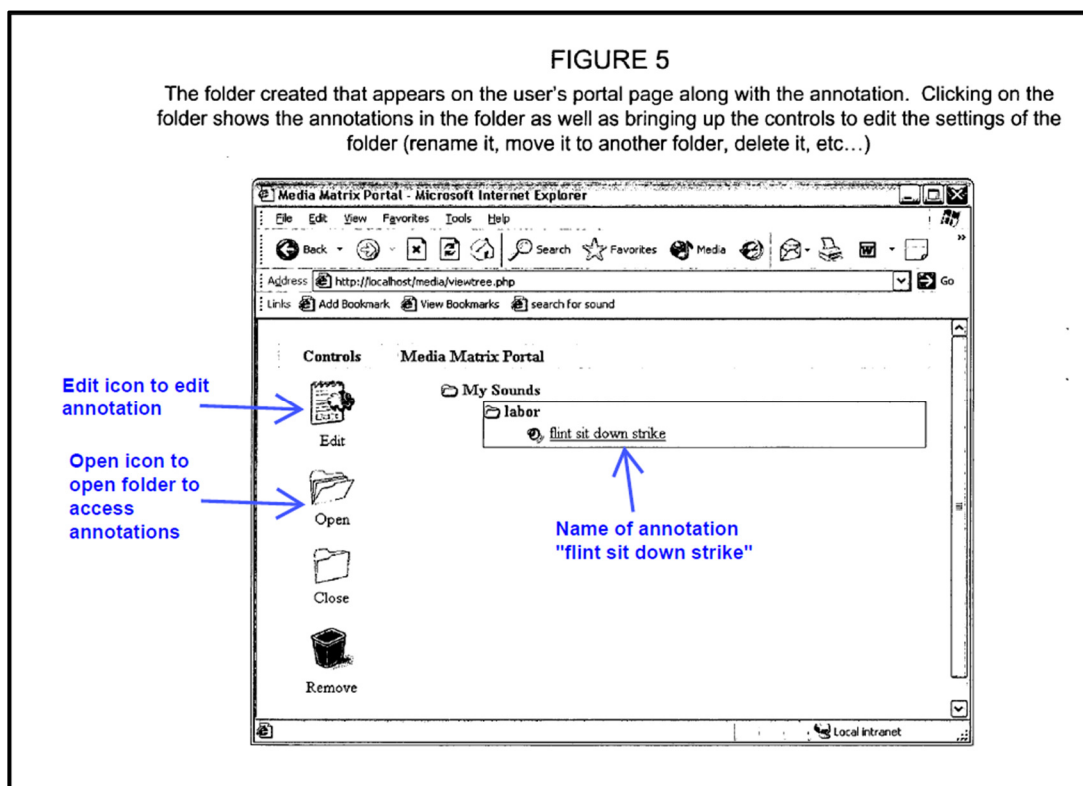
in Figure 5. Fegan, Fig. 5, [0081]. Figure 5 depicts a portal page window displaying a “user-created directory of annotative information associated with multimedia files,” which a POSITA would understand to be the segments/annotations generated by the client-side application, as discussed above for Figure 4. Fegan, [0081].

208. The portal page of Figure 5 includes an “edit” icon that the user can select to edit the annotation, as described below:

[0081] After a user designates a location for storage and retrieval of the annotative information [in Figure 4], the information appears on the user’s personal portal page that can be **accessed through a browser window**. FIG. 5 shows a **browser window** displaying a **user-created directory of annotative information** associated with multimedia files. In preferred embodiments of the present invention, the directory **allows a user to select, open, edit, close, or delete the annotative information** associated with a multimedia file.

Fegan, [0081] (emph. added).

209. The top of Figure 5 describes the window as “[t]he folder created that appears on the user’s portal page along with the annotation.” Fegan, Fig. 5 (“Clicking on the folder shows the annotations in the folder as well as bringing up the controls to edit the settings of the folder (rename it, move it to another folder, etc. . .”). Figure 5 is depicted below:



Fegan, Fig. 5 (annotated).

210. A POSITA would understand Figure 5 and paragraph [0009] as disclosing a portal page that provides access to the user's annotations stored at the portal location. The user can select the "Edit" icon to edit a selected annotation (e.g., "flint sit down strike") for editing as depicted in Figure 7. See Fegan, [0081], Fig. 7. A POSITA would understand the portal page of Figure 5 does not itself provide for editing of annotations, rather, it provides an icon link ("Edit") that can be selected to access another client-browser generated window for editing an annotation in Figure 7. Fegan, [0081], Figs. 5, 7. The Petitioner's argument based on paragraph [0009] and Figure 5 does not meet the claim limitation. See Pet., 35; Houh Decl., ¶

155. Accordingly, Figure 5 of Fegan does not disclose a server-side *video editing environment* application according to Limitation 13.1.

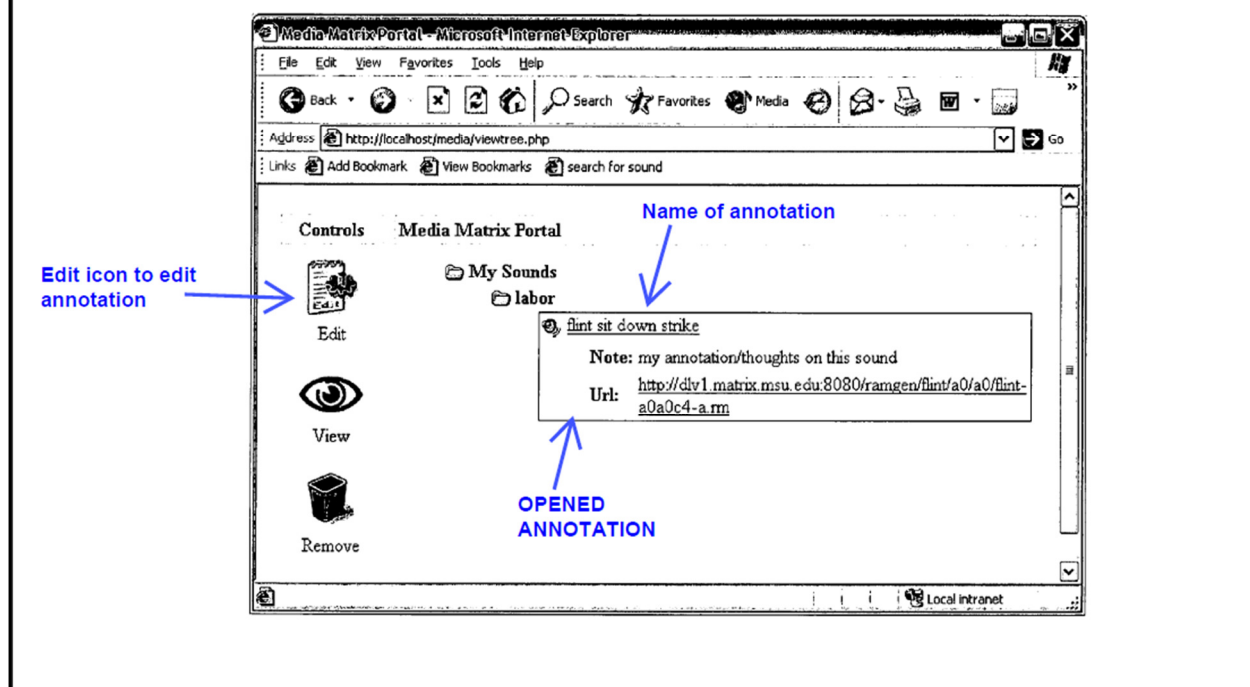
8. Fegan’s Discussion of the Annotation View Window of Figure 6 Does Not Disclose a Server-Side Video Editing Environment Application Per Limitation 13.1

211. Fegan discloses that when a control button is selected in Figure 5, an annotation view window is generated and displayed by the client-side software, as depicted in Figure 6. Figure 6 allows the user to “determine the contents of an annotation **prior to opening it for editing.**” Fegan, [0081]. Fegan describes Figure 6 as follows:

[0081] ... FIG. 6 shows a browser window in which saved annotative information has been selected from the directory shown in FIG. 5. When a user selects an annotation, **data comprising the annotative information is displayed in the browser window** [of Figure 6], **allowing the user to determine the contents of an annotation prior to opening it for editing.** FIG. 7 shows a saved annotation **that has been opened** via the directory for further editing. In preferred embodiments, annotations may be **freely created, edited, and deleted** without limitation.

Fegan, [0081] (emph. added). Fegan also states at the top of Figure 6: “Clicking on the annotation brings up it’s info as well as controls for working with the annotation (view, edit, delete, etc. . .)”. Figure 6 is depicted below:

FIGURE 6
Clicking on the annotation brings up it's info as well as controls for working with the annotation (view, edit, delete, etc...)

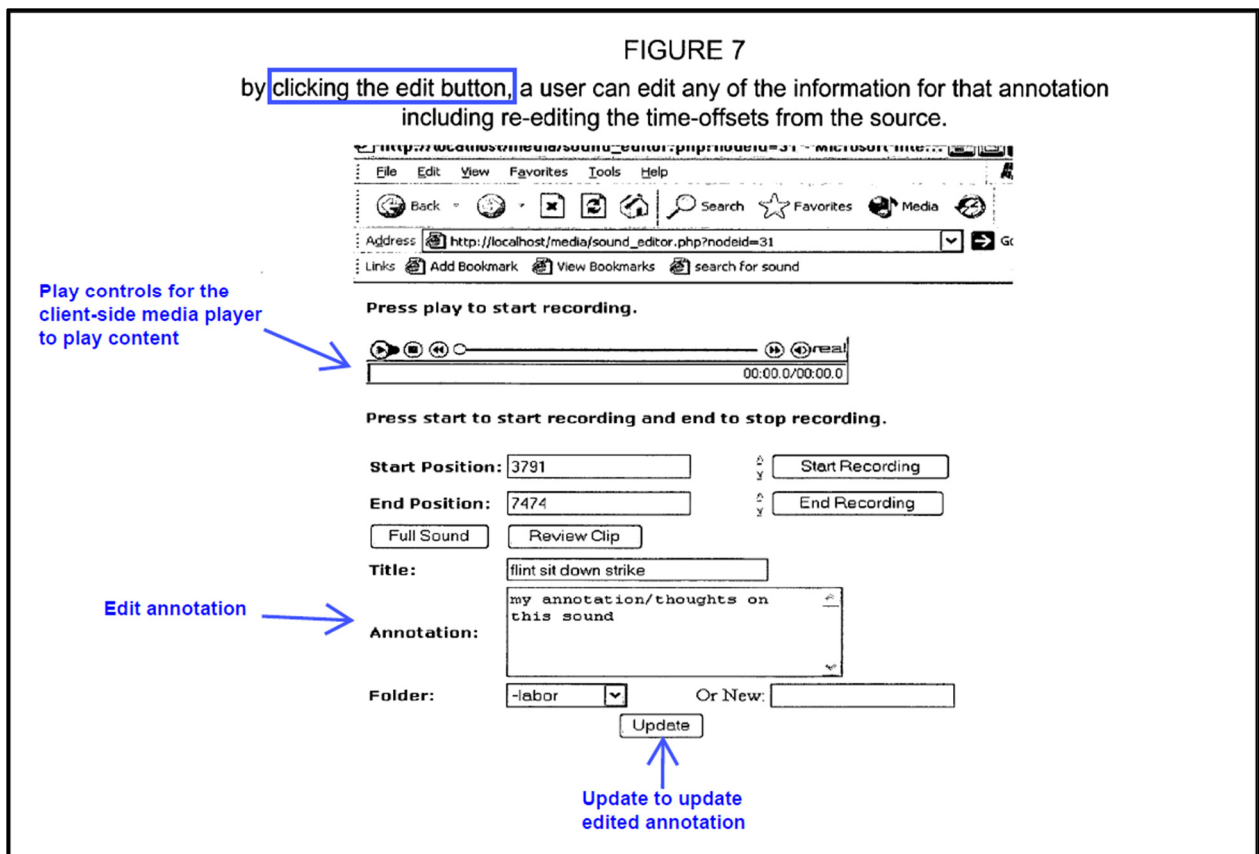


Fegan, Fig. 6 (annotated).

212. Accordingly, a POSITA would understand that Figure 6 is an annotation view window generated by client-side software. Fegan, [0081]. Fegan does not disclose in connection with Figure 6 a server-side *video editing environment* application as required by limitation 13.1. Additionally, Fegan describes Figure 6 as providing a software button (Edit icon) that can be selected to transition to a different browser page (Figure 7) to perform editing of annotations. A POSITA would understand that Figure 6 itself does not itself provide for editing of annotations as required by limitation 13.1. Thus, Fegan does not disclose limitation 13.1.

9. Fegan’s Discussion of the Segment/Annotation Edit Window of Figure 7 Does Not Disclose a Server-Side *Video Editing Environment* Application Per Limitation 13.1

213. Fegan discloses in Figure 7 “a saved annotation that has been opened via the directory for further editing.” Fegan, [0081]. Fegan states that preferably “annotations may be freely created, edited, and deleted without limitation.” Fegan, [0081]. The top of Figure 7 states that “by clicking the edit button [on Figure 5 or Figure 6], a user can edit any of the information for that annotation including re-editing the time-offsets from the source.” Fegan, Fig. 7. A POSITA would understand Figure 7 as depicting a segment/annotation edit window that allows a user to edit a retrieved annotation, as illustrated below:



Fegan, Fig. 7 (annotated).

214. A POSITA would understand Fegan discloses Figure 7 as a browser window for editing annotations using **client-side software** (browser, media player). See Fegan [0074] (“software directs **the browser to load the file** into a **browser based multimedia player**”). Nowhere does Fegan disclose any server-side application performing the editing of annotations in connection with Figure 7. Accordingly, Figure 7 does not disclose a **server-side video editing environment** application per limitations 13.1-13.5.

10. Summary of Fegan Regarding Limitation 13.1 for a Server-Side Video Editing Environment Application

215. I have analyzed each of Figures 1-7 of Fegan and the accompanying disclosure. Fegan does not disclose a server-side *video editing environment* application enabling remote cue point editing at a client terminal per limitation 13.1. I further note that the Petition makes no inherency arguments for any of its theories for limitation 13.1.

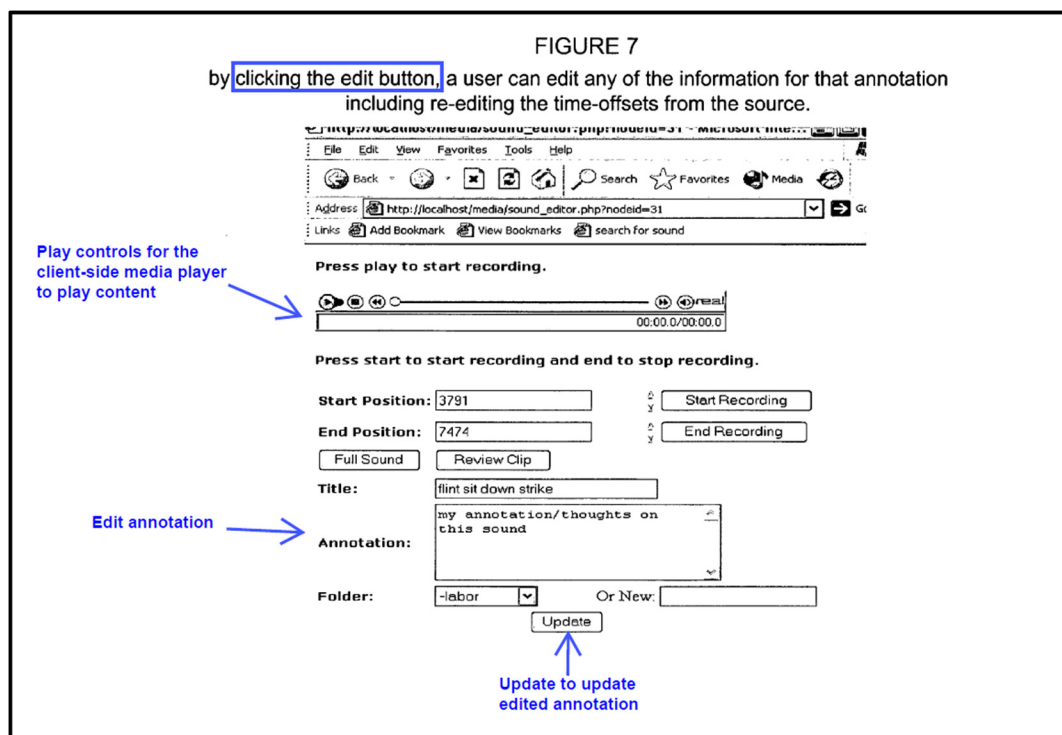
C. Fegan Fails to Disclose *Receiving a Cue Point Editing Command ... From a Remote Client Terminal And Modifying a Portion of the Cue Point Information ... According to the Cue Point Editing Command* (Limitations 13.4-13.5)

216. Limitations 13.4-13.5 of Claim 13 recite in a straightforward fashion that the cue point information at the *video editing environment* is modified in accordance with the *cue point editing command* received from the client terminal.

217. In its Institution Decision for the -00710 IPR, the Board raised a question as to whether Fegan discloses Limitations 13.4-13.5. See Ex. 2034 (Decision on Institution, Paper 13, 9/5/2024), 33 (calling into question “whether Fegan discloses Limitations 13.1, 13.4, 13.5, and Claim 15”).

1. Limitation 13.4 for *Receiving a Cue Point Editing Command*

218. The Petitioner relies on Figure 7 of Fegan, which displays the segment/annotation edit window enabling a user to edit a segment of multimedia content and an associated annotation. See Section IX.B.8-B.9. Fegan, [0081], Fig. 7. Figure 7 is illustrated below:



Fegan, Fig. 7 (annotated).

219. The Petitioner asserts that the user's selection of "Update" in Figure 7 causes the client to transmit a "command to edit the annotation" that is "sent from the client computer and received using the computer system's remote server" to allegedly meet limitation 13.4. Pet., 45. But Fegan only states the following regarding Figure 7:

[0081] FIG. 7 shows a saved annotation that has been opened via the directory for further editing. In preferred embodiments, annotations may be freely created, edited, and deleted without limitation.

Fegan, [0081].

220. Nowhere does Fegan disclose that selecting the "Update" button in Figure 7 causes the client to transmit a command to the server-side, as asserted. Pet., 45. Indeed, Fegan does not provide **any disclosure** about what operations or events are triggered by the user selecting "Update." The Petitioner's and its expert's citation to various paragraphs that describe that a user can edit annotation information or store annotation information (see Pet., 45-46, citing to Fegan, [0003], [0009], [0037], [0081], [0083], [0085]) does not remotely demonstrate Fegan to be disclosing that a user selecting the "Update" button causes the client to transmit a *cue point editing command* received at the server side per limitation 13.4. See Houh Decl., ¶¶ 169-177.

221. The Petitioner makes no argument that it is inherent that selecting the "Update" button causes the client to transmit a command over the network to the

server-side. See Pet., 43-46; Houh Decl., ¶¶ 169-177. At most, a POSITA would understand that selecting the “Update” button on Figure 7 could cause the edited segment/annotation information to be locally updated on the client device executing the client-side software as I have previously discussed. I have already demonstrated that the windows of Figures 4-7 are generated based on **client-side software** (client browser, media player), and thus it would be logical that updated annotation information would be locally updated on the client rather than a command being sent to the server-side per limitation 13.4. See Section IX.B. Accordingly, it is not inherent that selection of the “Update” button causes a command to be sent from the client to the server in accordance with limitation 13.4.

222. Accordingly, Fegan does not disclose Limitation 13.4, and thus Claim 13 is not anticipated.

2. Limitation 13.5 for *Modifying ... the Cue Point Information ... According to the Cue Point Editing Command ... From the Client Terminal*

223. The Petitioner’s argument for limitation 13.5 parallels its argument for limitation 13.4. See Pet., 46-47. Houh Decl., ¶¶ 178-179. Figure 7 of Fegan does not disclose that edited annotation information is modified at the server-side in response to a command transmitted from the client in accordance with limitation 13.5. Accordingly, Fegan does not disclose Limitation 13.5, and thus Claim 13 is not anticipated.

D. Fegan Fails to Disclose Claim 15 for Receiving an Edit Mode Entry Signal ... From the Client Terminal Representing Entry into the Cue Point Edit Mode, and Switching the Video Editing Environment to the Cue Point Editing Mode in Response to Receiving the Edit Mode Entry Signal

224. Claim 15 recites an “*edit mode entry signal*” sent over the network from the client to the server-side that is separate from the “*cue point editing command*” of limitation 13.4 is that is sent from the client over the network. The “*VEE switch[es] ... to the cue point editing mode in response to receiving the edit mode entry signal.*” Thus, Claim 15 calls for two separate signals sent from the client over the network to the server-side application: (1) an *edit mode entry signal* and (2) the *cue point editing command*.

225. This understanding of the *edit mode entry signal* is supported by the claim language and specification of the ’286 Patent, as discussed above regarding Gupta in Section VIII.G.

226. In its Institution Decision for the -00710 IPR, the Board raised a question as to whether Fegan discloses Claim 15. See Ex. 2034 (Decision on Institution, Paper 13, 9/5/2024), 33 (calling into question “whether Fegan discloses Limitations 13.1, 13.4, 13.5, and Claim 15”).

227. The Petition asserts that the *cue point editing mode* is entered when the client selects the “Edit” button in portal window of Figure 5, which results in the

display of the segment/annotation edit window of Figure 7. Pet., 49-50. See Section IX.B.7-B.9.

228. However, there is no evidence showing that selecting the “Edit” button on Figure 5 causes the transmission of a command signal from the client over the network to the server side in Fegan. Fegan only states the following regarding the Edit button on Figure 5:

[0081] After a user designates a location for storage and retrieval of the annotative information [in Fig. 4], the information appears on the user’s personal portal page [in Fig. 5] that can be accessed through a browser window. FIG. 5 shows a browser window displaying a user-created directory of annotative information associated with multimedia files. In preferred embodiments of the present invention, the directory allows a user to select, open, **edit**, close, or delete the annotative information associated with a multimedia file. FIG. 6 shows a browser window in which saved annotative information has been selected from the directory shown in FIG. 5. When a user selects an annotation, data comprising the annotative information is displayed in the browser window [in Fig. 6], allowing the user to determine the contents of an annotation prior to opening it for editing. FIG. 7 shows a saved annotation that has been opened via the directory for further editing. In preferred embodiments, annotations may be freely created, **edited**, and deleted without limitation.

Fegan, [0081] (emph. added).

229. Accordingly, Fegan does not disclose that selecting the “Edit” button in Figure 5 (or Figure 6) causes the client to transmit a command over the network to the server side, as required by Claim 15. Fegan’s disclosure and Figures 5-6 only indicate that selecting the “Edit” button causes the segment/annotation edit window of Figure 7 to be displayed. See Fegan, [0081], Figs. 5-7. See Section IX.B.7-B.9.

230. The Petitioner and its expert do not point to any evidence that the user selecting the “Edit” button causes the client to transmit a command over the network to the server-side. See Pet., 50; Houh Decl., ¶¶ 188-199. A POSITA would understand that the fact that Fegan uses HTTP to communicate does not disclose the “*edit mode entry signal*” of Claim 15. For that reason alone, a POSITA would understand that the *edit mode entry signal* of Claim 15 is not disclosed by Fegan.

231. The Petitioner does not make an inherency argument regarding the “Edit” button. See Pet., 49-50; Houh Decl., ¶¶ 188-199. Further, a POSITA would understand that selecting the “Edit” button causes the client-side software (browser, media player) to generate the segment/annotation edit window of Figure 7, that is the alleged *cue point editing mode*. Pet., 49-50. See Section IX.B.6-B.9 (Figs. 4-7). Thus, selecting the “Edit” button causes the client-side software to generate the new window of Figure 7, not send a command over the network to the server-side as required by Claim 15. This demonstrates that Fegan does not disclose the claimed *edit mode entry signal* expressly or inherently.

232. For the above reasons, Claim 15 is not anticipated by Fegan.

E. Fegan Fails to Disclose Claim 18 for *Receiving, Over the Network, Authentication Information Related to Access Authority of the Remote Client Terminal*

233. Anticipation requires that Fegan must disclose, either expressly or through inherency, that the server side receives “*over the network, authentication information related to access authority*” of the client terminal. Fegan does not disclose that limitation.

234. Petitioner and its expert cite Fegan’s reference to the user having a “personal portal hosted on the system” that “allows the user to edit and organize collected materials.” Pet., 51 (citing Fegan, [0009], [0081]); Houh Decl., ¶¶ 200-203. But those passages do not disclose receiving “*authentication information relating to access authority over the network*” at the server-side per Claim 18.

235. The Petition (at 51) also cites to the generic definition in Fegan for “security protocol” as being:

[0050] As used herein, the term “security protocol” refers to an electronic security system (e.g., hardware and/or software) to limit access to processor to specific users authorized to access the processor. For example, a security protocol may comprise a software program that locks out one or more functions of a processor until an appropriate password is entered.

Fegan, [0050]. Houh Decl., ¶¶ 200-203. This passage does not disclose the client terminal sending “*authentication information relating to access authority **over the network***” to the server-side per Claim 18.

236. The Petitioner has made no inherency argument regarding Claim 18. See Pet., 50-51; Houh Decl., ¶¶ 200-203. Furthermore, as discussed for Gutpa in Section X.H, there are multiple other mechanisms that could be used in Fegan for security information to be entered other than by transmitting authentication information to the server over the network. Accordingly, Fegan does not inherently disclose the additional limitation of Claim 18.

F. Claims 14-15 and 18-20 Are Patentable Over Fegan

237. Claims 14-15 and 18-20 are patentable over Fegan for at least the same reasons set forth for Claim 13.

X. CLAIM 17 IS PATENTABLE OVER FEGAN AND SULL (GROUND 2* FROM -01101 IPR)

238. Regarding Fegan, the Petitioner challenges Claim 17 as obvious based on Fegan and Sull. See Pet. (IPR2024-01101), 1. Sull is relied upon only for allegedly teaching the automatic generation of cue point information. Pet. (IPR2024-01101), 53-55. The Petitioner does not rely on Sull for any of the limitations of independent Claim 13. Pet. (IPR2024-01101), 37-53. The Petitioner does not rely on Fegan for the additional limitation of Claim 17. The Petitioner relies only on Fegan for the limitations of independent Claim 13. Pet. (IPR2024-01101), 37-53.

Accordingly, Claim 17 is patentable over Fegan and Sull for at least the reasons set forth in Section IX above.

XI. CLAIMS 13-15 AND 17-20 ARE PATENTABLE OVER KIM (GROUND 3)

A. Claims 13-15 and 17-20 Are Patentable Over Kim

239. Kim fails to teach or suggest the following limitations of Claim 13:

- *“providing a computer system ... comprising a video editing environment, and being disposed at a server side of a public network”* (Limitation 13.1); and
- *“receiving a cue point editing command over the network from a remote client terminal”* and *“modifying a portion of the cue point information at the server side ... according to the cue point editing command”* (Limitations 13.4-13.5).

B. Kim Fails to Meet Limitation 13.1 for *Providing a Computer System ... Comprising a Video Editing Environment, and Being Disposed at a Server Side of a Public Network*

240. In its Institution Decision in the -00710 IPR, the Board raised the question of whether Kim teaches Limitation 13.1. See Ex. 2034 (Decision on Institution, Paper 13, 9/5/2024), 35-36 (calling into question “whether Kim discloses Limitations 13.1, 13.4, 13.5, and Claim 15”). The Board invited the parties to address Kim’s annotation and editing system, particularly the multimedia player and Multimedia Bookmark BBS Server. *Id.*, 36.

241. As discussed above in Section VIII.C, the inventor repeatedly emphasized during the prosecution of the '286 Patent the *video editing environment* as a server-side application for performing editing of cue point information that enables **remote cue point editing** by the client terminal as opposed to local cue point editing using a local client application. This understanding is supported by the '286 Patent specification and the language of Claim 13, also discussed in Section VIII.C.

242. Kim discloses a system allowing clients to create multimedia bookmarks (also referred to as “VMarks”) associated with corresponding multimedia content files. The locally-created bookmarks are locally-stored at the user’s client device and, if the user elects, can be uploaded to a Multimedia Bookmark BBS server so that users can share bookmarks. Ex. 1006 (Kim), Abstract, Figs. 1-2.

243. As disclosed in Kim, the bookmarks are created locally using the **client side multimedia bookmark client**, and they are stored locally in the client bookmark storage. Kim, Abstract (“a client comprising storage for VMark, a web browser, a media player and a Vmark client.”), Fig. 2 (multimedia bookmark client 238, multimedia bookmark storage 236), Fig. 20A (multimedia bookmark client 2030, client storage 2032), [0211] (VMark client), [0283] (“multimedia bookmark client 238 located in client 230”), [0288] (“The multimedia bookmark saved at local storage 236 of client 230 ... can be uploaded to multimedia bookmark BBS server

212”). Kim teaches that the client device has client-side software including “a media player, and a VMark client.” Kim, [0211]. The VMarks are created at the client by the user. Kim, [0215] (“at a first client, making a VMark”), [0275]-[0281] (client-side player window 102 with client-side multimedia player 104 “for making a multimedia bookmark” including add bookmark button 122 and post bookmark to BBS button 132), Figs. 1-2. Kim explains that the add-bookmark button is used by a viewer using the client-side software of Figure 1 to add a bookmark to local persistent storage, and the post bookmark to BBS button 132 is used to post the bookmark to the BBS server 112. Kim, [0281]. This confirms that the bookmarks are locally created on the client. Kim also discusses the ability of a client user to “share” bookmarks via the bookmark server, which confirms the bookmarks are locally created at the client. Kim, [0111]. Thus, Kim teaches a **client-based system for local bookmark editing**, where the creation of bookmarks takes place at a local multimedia bookmark client 238 running on the client device 230. The Petitioner and its expert admit this to be the case, stating: “[M]ultimedia BBS server 212 receives a multimedia bookmark **created locally at client 230.**” Pet., 57; Houh Decl., ¶ 236 (“the multimedia bookmark BBS server 212 receives a multimedia bookmark that had been **created locally at client 230**”) (all emph. added).

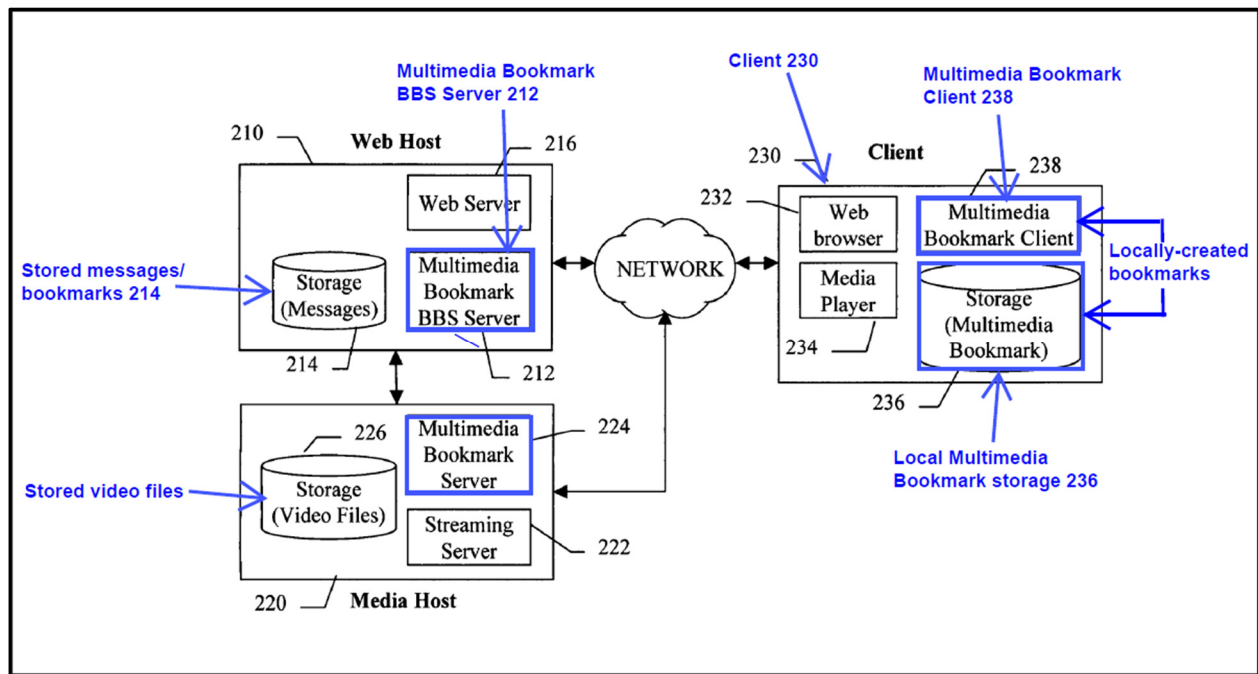
244. Kim does not teach or suggest the server-side *video editing environment* application as claimed. Kim instead provides for local bookmark editing, not remote

bookmark editing provided by a server-side *video editing environment* application per limitation 13.1. I provide further analysis supporting this conclusion below.

1. Kim's Bookmarks Are Created Locally Using the Multimedia Bookmark Client 238 Running on Client 230

245. As discussed above, a POSITA would understand that Kim describes that bookmarks associated with segments of multimedia content such as video are created locally by the multimedia bookmark client 238 that is executed by the client 230. The locally-created bookmarks are stored in the client's local storage 236.

Figure 2 illustrates these operations:



Kim, Fig. 2 (annotated).

246. Kim describes that the client executes client-side applications including media player 234 (for playing streaming media) and the multimark bookmark client

238 (for creating bookmarks that are locally-stored in memory 236 and for retrieving them from local memory 236 for upload to the multimedia bookmark BBS server 212). Kim, [0284]. The multimedia bookmark client may be embedded in the web browser. Kim, [0297].

247. Kim describes the process of locally-creating bookmarks by multimedia bookmark client 238 running on client 230 and storing the bookmarks in local bookmark storage 236 at the client:

[0285] While a **user in client 230** is watching a video streamed from media host 220, the user can conveniently bookmark any position of the video and wave the multimedia bookmark **into the user's local storage 236** by clicking add-bookmark control button 108 in the player shown in FIG. 1.

[0287] The **client 238, which is application software** responsible for interactions between web browser 232 and local storage 236, **stores the multimedia bookmark image into local storage 236 together with the bookmark information such as video URI, start time, duration and etc.** The bookmark client 238 is also used to load the multimedia bookmark saved at the local storage into the web browser, so that the web browser can display the multimedia bookmark image and its information.

[0344] **Client 2026 comprises media player 2028 and multimedia bookmark client 2030 that takes charge of making and managing multimedia bookmarks stored at local storage 2032.**

See Kim, [0285], [0287], [0344], Fig. 2 (multimedia bookmark client 238 on client 230, local bookmark storage 236 on client 230), Fig. 20A (multimedia bookmark client 2030 on client 2026, local bookmark storage 2032 on client 2026) (all emph. added).

248. As noted above in paragraph [0287], the multimedia bookmark may include location information, such as the URI of the multimedia file and the position of the segment (start time or location, etc.). The bookmark may also include content information, such as textual annotations to the segment, bookmark title, a bookmark ID, and an image captured from the segment (e.g., the first frame of the bookmarked segment). Kim, [0264]-[0272]. The image is acquired by the multimedia bookmark client 238 making a request through the web browser 232 for the multimedia bookmark server 224 to capture the image and provide it back to the multimedia bookmark client 238 so that it can be stored into local storage with the other bookmark information. Kim, [0286]. Alternatively, the client-side multimedia bookmark client 238 can capture the image itself and store it with the rest of the bookmark information. Kim, [0287]. A POSITA would understand that, in either case, the client-side application is creating the bookmark by extracting and assembling the various pieces of information making up the bookmark.

249. Elsewhere, Kim confirms that the creation of bookmarks is performed client-side: “The client 238 ... stores the multimedia bookmark image into local

storage 236 together with the bookmark information such as video URI, start time, duration and etc.” Kim, [0287].

250. Addressing the Board’s questions (Instn. Dec., 36), the creation and editing of bookmarks in Kim occurs at the client using client-side software including the multimedia player. The Multimedia Bookmark BBS Server functions to receive bookmarks locally-created by the client 230 for bookmarks the user elects to share with other users. Kim, [0288] (“multimedia bookmark saved at local storage 236 of client 230 ... can be uploaded to multimedia bookmark BBS server 212 ... and then stored at storage 214 of the web host so that other users can share the multimedia bookmark.”), [0290] (“the ‘write’ (post or upload) control button 318 is selected when a user wants to post a multimedia bookmark” to the Multimedia Bookmark server).

251. Accordingly, a POSITA would understand that Kim teaches **local bookmark editing** using the **client-side** multimedia bookmark client 238 (and the client-side web browser 232 and client-side media player 234, as needed). Kim does not remotely teach or suggest a **server-side** *video editing environment* application for editing bookmarks to provide **remote bookmark editing** by the client terminal as per limitation 13.1. Moreover, Kim does not teach or suggest such a server-side application that causes the server-side editing of bookmarks as further set forth in limitations 13.4-13.5.

2. Kim's Server-Side Components Do Not Provide a Server-Side *Video Editing Environment* per Limitation 13.1

252. A POSITA would understand that the server-side components of Kim, including the Web Host and the Media Host, do not provide a server-side *video editing environment* application for bookmark editing to enable remote bookmark editing by the client device 230.

253. The Web Host has a web server 216 for interfacing with the client's web browser 232; a Web storage 214 for storing bookmark messages; and a multimedia bookmark BBS server 212 for receiving and storing locally-created bookmarks that the user has selected for upload to the BBS to be shared with other users. The Media Host has a multimedia bookmark server 224 for servicing bookmark requests; a streaming server 222 for streaming video to a client 230; and a storage 226 of video files to be streamed to a user at a client 230. Kim describes the server-side components as follows:

[0283] FIG. 2 illustrates the general system architecture of the multimedia bookmark BBS, according to an embodiment of the present disclosure. The system comprises multimedia bookmark BBS server 212 located in web host 210, multimedia bookmark server 224 located in media host 220 and multimedia bookmark client 238 located in client 230. The web host 210, media host 220 and client 230 are connected by a conventional communication network ("NETWORK").

[0284] Web host 210 provides lists of videos which have corresponding links or URIs of the Videos stored at media host 220 with a hypertext

markup language (HTML). Media host 220 stores the videos in its local storage 226, and provides them to client 230 when they are requested. Client 230 can select one of the videos from the lists displayed in web browser 232. The selection process requests the video to be serviced to media host 220, and client 230 receives the video streamed by streaming server 222 of media host 220 and displays the video on media player 234.

[0285] While a user in client 230 is watching [using media player 234] a video streamed from media host 220, the user can conveniently bookmark any position of the video and save the multimedia bookmark into the user's **local storage 236** by clicking add-bookmark control button 108 in the player shown in FIG. 1.

[0288] The **multimedia bookmark saved at local storage 236 of client 230 ... can be uploaded to multimedia bookmark BBS server 212** of web host 210 and then stored at storage 214 of the web host so that other users can share the multimedia bookmark. Thus, one who retrieves the bookmark from multimedia bookmark BBS server 212 can start to play the video exactly from the bookmarked position.

[0289] Media host 220 comprises streaming server 222, bookmark server 224 and storage 226 for archiving media files. Bookmark server 224 is responsible for handling the request from bookmark BBS server 212. The multimedia bookmark server [224] obtains a bookmark image at the required position in accordance with the request and then sends the captured bookmark image to multimedia bookmark BBS server 212 as a reply for the request of multimedia bookmark image. Streaming server 222 is responsible for the request from client 230 to play a video.

Kim, [0283]-[0285], [0288]-[0289] (emph. added).

254. In fact, the Petitioner inadvertently concedes that Kim provides for the client-side creation and editing of bookmarks. Pet., 62 (“A POSITA would have looked at Kim’s [] client-server architecture and understood that adding and managing bookmarks is the function of client 230.”).

255. As illustrated from the above passages, Kim does not teach or suggest a server-side *video editing environment* application for performing bookmark editing to enable **remote bookmark editing** by the client terminal, as per limitation 13.1. As discussed above in Section XI.B.1, Kim instead teaches a **client-based solution** based on the client-side multimedia bookmark client 238 running on client 230 that provides for **local bookmark editing**.

3. Petitioner Aylo’s Arguments Fail to Establish Kim Meets Limitation 13.1 for a Server-Side *Video Editing Environment*

256. The Petitioner and its expert cite miscellaneous passages from Kim to assert that Limitation 13.1 is satisfied. Pet., 54-56; Houh Decl., ¶¶ 220-228. None of these passages teach or suggest the server-side *video editing environment* application per Limitation 13.1.

257. I have examined the assertions made by Dr. Houh that Kim teaches a server-side *video editing environment* application, and find them not to be credible.

258. The Petitioner and its expert assert that multimedia bookmark BBS server 212 in Web Host 210 (with or without Media host 220) “executes software for editing bookmarks.” Pet., 55; Houh Decl., ¶ 224. There is no evidence in Kim to support this assertion, and the Petitioner and its expert cite no passages in Kim referencing “software for editing bookmarks.”

259. For example, the Petitioner and its expert reference the multimedia bookmark server 224 that is “running at media host 220.” Pet., 54-55 (citing Kim, [0326]); Houh Decl., ¶ 222. The cited passage at paragraph [0326] of Kim merely describes that multimedia bookmark server 224 can capture an image from stored video and send the image to the client. Kim, [0326]. See Pet., 54; Houh Decl., ¶ 222. For the same reason, the description at paragraph [0286] of multimedia bookmark server 224 obtaining a bookmark image and providing it to the multimedia bookmark client 238 does not teach or suggest the server-side *video editing environment* application for **editing** bookmarks per limitation 13.1. Pet., 55 (citing Kim, [0286]); Houh Decl., ¶¶ 224-225 (citing Kim, [0286], [0345]). Kim’s statement (at [0286]) that the “bookmark BBS server 212 sends the multimedia bookmark image together with the multimedia bookmark information to the web browser 232 of client 230” (see Houh Decl., ¶ 224) does not teach or suggest the server-side **editing** of bookmarks per limitation 13.1. The bookmarks are created by the client-side software, as discussed in Section XI.B.1.

260. The Petitioner and its expert cites to the local multimedia bookmark client 230 uploading a locally-created bookmark from local storage 236 to the multimedia bookmark server 212. Pet., 55 (citing Kim, [0288]); Houh Decl., ¶ 224. The uploading of a locally-created bookmark from the client 230 does not correspond to a server-side *video editing environment* for **editing** bookmarks per limitation 13.1.

261. The Petitioner asserts that “[s]treaming server 222 executes software for streaming video.” Pet., 55 (citing Kim, [0284], [0289]). Even if Kim provided that disclosure (it does not), it does not teach or suggest that streaming server 222 provides a server-side *video editing environment* application for **editing** bookmarks per limitation 13.1. See Houh Decl., ¶ 226.

262. The Petitioner and its expert refer to an embodiment in Kim where a locally-created bookmark can be emailed from one user to another user, and where the system automatically attaches bookmark advertisements to the email, as illustrated in Figure 17 of Kim. Kim, [0033] (“When a user sends a multimedia bookmark e-mail to others, a multimedia bookmark e-mail system can attach advertising multimedia bookmarks to the user’s multimedia bookmark e-mail automatically.”), Fig. 17. See Pet., 55; Houh Decl., ¶ 224. In connection with this embodiment, Kim describes that an administrator can “make[] his/her own multimedia bookmarks ... and then select[] some multimedia bookmarks to be

attached from his/her own multimedia bookmarks.” Kim, [0334]. The administrator can “edit video title or allowable playing duration” of the advertisement bookmarks to be attached to the user’s email sending his/her bookmark to another user. Kim, [0335], Figs. 18B-C.

263. Kim does not teach or suggest that the editing of advertisement bookmarks is performed by a server-side *video editing environment* application providing for bookmark editing to enable remote bookmark editing per limitation 13.1. Kim does not disclose any application on the Web Host 210 or the Media Host 220 for performing the editing of advertisement bookmarks for the bookmark advertising embodiment. See Kim, [0333]-[0336]. Accordingly, Kim does not teach or suggest a server-side *video editing environment* application for editing bookmarks to enable remote bookmark editing by the client per limitation 13.1.

264. Instead, a POSITA would understand from Kim that the editing of bookmarks in the bookmark advertising embodiment would be performed using a client with administrator access rights using client-side applications such as client 230 running the multimedia bookmark client 238, as previously described, which provides for **local bookmark editing**, not **remote bookmark editing** using a server-side *video editing environment* as claimed. See Section XI.B.1-B.2. In fact, elsewhere the Petitioner acknowledges that any editing of bookmarks in the

bookmark advertising embodiment would be performed by an administrator using client 230. Pet., 62, 66 (“an administrator at client 230”).

265. Accordingly, Kim fails to teach or suggest a *video editing environment* application for performing editing of bookmarks to provide remote bookmark editing for a client terminal per limitation 13.1.

C. Kim Fails to Meet *Receiving a Cue Point Editing Command ... From a Remote Client Terminal And Modifying a Portion of the Cue Point Information ... According to the Cue Point Editing Command* (Limitations 13.4-13.5)

266. Limitations 13.4-13.5 of Claim 13 provide that the cue point information at the *video editing environment* is modified in accordance with the *cue point editing command* received “*over the network*” from the client terminal.

267. In its Institution Decision in the -00710 IPR, the Board raised a question as to whether Kim discloses Limitations 13.4-13.5. See Ex. 2034 (Decision on Institution, Paper 13, 9/5/2024), 36 (calling into question “whether Kim discloses Limitations 13.1, 13.4, 13.5, and Claim 15”).

1. Limitation 13.4 for *Receiving a Cue Point Editing Command Over the Network from a Remote Client Terminal*

268. I believe that the Petitioner understands that Kim is deficient regarding limitations 13.4-13.5, which is why the Petitioner sets forth back-up Ground 4 which relies on Gupta to provide the *cue point editing command* that is not taught by Kim. See Pet., 71-72.

269. For this Ground 3 based on Kim, the Petitioner points to various control buttons on Kim's client-side user interfaces as allegedly teaching the claimed *cue point editing command* received “*over the network*” from a *remote client terminal*. However, each of the control buttons simply transitions the user from one local user interface to another local user interface at the client. There is no teaching or suggestion in Kim that selecting any of these control buttons causes the client to send a *cue point editing command* “*over the network*” to be received at the server-side per limitation 13.4.

270. Figures 17 and 18A-18C of Kim discuss the bookmark advertising embodiment where the system automatically adds advertising bookmarks to an email a first user is sending to a second user with the first user's bookmark. Kim, [0333]-[0336], Figs. 17, 18A-18C. The administrator can select the advertising bookmarks to be appended to the message. Kim, [0334]-[0336]. The Petitioner points to the administrator selecting the “Save” button 1822 in Figure 18A, selecting the “Save” button 1838 in Figure 18B, or selecting the “Save” button 1850 in Figure 18C, which are user interfaces enabling the administrator to select and preview the advertising bookmarks to be appended to the email. Pet., 58-59 (citing Kim, [0334]-[0335]). Houh Decl, ¶¶ 241-244.

271. Kim describes the Save button 1822. “The administrator then completes the selection [of advertisements] by clicking on save button 1822, and a new GUI

screen 1830 of FIG. 18B will appear.” Kim, [0334]. Thus, selecting the Save button 1822 causes the user interface to change from the advertisement selection window of Figure 18A to the advertisement selection window of Figure 18B. There is no teaching or suggestion that selection of the Save button 1822 causes the client 230 to send a *cue point editing command* “over the network” to the server-side, as required by limitation 13.4.

272. Kim describes the Save button 1838. “The administrator then completes the verification by clicking on Save button 1838, and a new GUI screen 1840 of FIG. 18C will appear.” Kim, [0335]. Thus, selecting the Save button 1838 causes the user interface to change from the advertisement selection window of Figure 18B to the advertisement preview window of Figure 18C. Kim, [0335], Figs. 18B-18C. There is no teaching or suggestion that selection of the Save button 1838 causes the client 230 to send a *cue point editing command* “over the network” to the server-side, as required by limitation 13.4.

273. The Petition also refers to the “Save” button 1850 in the advertisement preview window of Figure 18C. Pet., 59 (citing Kim, [0335], [0336], [0314], [0299], [0315]). However, Kim describes the Save button 1850 as follows:

[0336] And then, **by clicking the save button 1850** with “Advertising Multimedia Bookmark Mail” radio button 1846 checked, the **advertising multimedia bookmarks are stored on database** to be **automatically attached into user’s multimedia bookmark e-mail**

whenever a user sends a multimedia bookmark e-mail to other. By clicking the save button 1850 with “Advertising News Letter” radio button 1848 checked, the advertising multimedia bookmarks are stored on database to be automatically attached into a news letter e-mail whenever a promotion is done by sending a news letter e-mail to users.

Kim, [0336] (emph. added). Accordingly, selecting the Save button 1850 in Figure 18C causes the selected advertisement bookmarks to be placed in storage and automatically attached to an email or newsletter. There is no teaching or suggestion that selecting the Save button 1850 causes the client 230 to send a *cue point editing command* “over the network” to the server-side, as required by limitation 13.4.

274. The Petition also refers to the “add-bookmark-title control button 122” used to “input a title of [a] bookmark given by a user” in Figure 1 of Kim, also described in the Sull reference as the “add-bookmark-title control button 922” for Figure 9 of Sull. See Kim, [0276], Fig. 1; Ex. 1007 (Sull), [0206], Fig. 9. See Pet., 58, 60-61 (citing Sull, [0055], [0051], [0174], [0205]-[0206], [0213]). See Houh Decl, ¶¶ 245-248. There is no teaching or suggestion in Kim or the cited sections of Sull that selecting the add-bookmark-title control button causes the client 230 to send a *cue point editing command* “over the network” to the server-side, as required by limitation 13.4. As explained in Section XI.B, the creation of bookmarks is carried out by client-side software (multimedia bookmark client 238 with local bookmark

storage 236) executed on client 230, not by server-side applications. Houh Decl., ¶¶ 245-251.

275. Dr. Houh exposes his reluctance on the argument for limitation 13.4 because he only asserts that clicking “save” in Kim or “add-bookmarking-title” in Sull results in “sending ... an instruction to the computer system,” but not sending an instruction “*over the network*” as claimed. Houh Decl., ¶ 251.

276. The Petition has a back-up argument that “a POSITA would have found it obvious from Kim’s and (Sull’s) teachings to use client 230 to send the editing command.” Pet., 61-62. Houh Decl., ¶ 252. In other words, a POSITA would modify Kim’s system, which uses client-side applications to create/edit annotations, to send commands to the server-side to create/edit annotations. Respectfully, that does not make sense, given that Kim is premised on the client-side creation/editing of annotations. The Petitioner then reasons that “it would be obvious to place bookmark editing functionality at client 230 as editing bookmarks is similar to functions already handled by client 230.” Pet., 62. The Petitioner’s expert repeats that assertion *verbatim*. Houh Decl., ¶ 252. However, this reasoning does not make sense because, as explained in Section XI.B for limitation 13.1, the Kim system already relies on a **local bookmark processing** using the **client-side multimedia bookmark client 238** to **locally create** bookmarks, instead of **remote bookmark processing** based on command signals from the client. Any bookmark **editing** operations in Kim are

locally-performed using the client-side application software running on client 230, and thus there would be no reason for the client to send a “*cue point editing command*” “*over the network*” to have editing performed on the server-side, as proposed.

277. Accordingly, it is my opinion that Kim does not meet limitation 13.4 for “*receiving a cue point editing command over the network from a remote client terminal.*” As set forth above, I disagree with the Petitioner’s and its expert’s argument. See Pet., 58-62; Houh Decl., ¶¶ 239-253.

2. Limitation 13.5 for *Modifying a Portion of the Cue Point Information at the Server Side ... According to the Cue Point Editing Command ... From the Client Terminal*

278. The Petition’s argument for limitation 13.5 parallels its argument for Limitation 13.4. See Pet., 62-63. Houh Decl., ¶¶ 254-256. Kim does not teach or suggest modifying bookmark information in response to a bookmark *editing command* sent from the client “*over the network*” to the server-side. Accordingly, Kim does not meet Limitation 13.5.

D. Kim Fails to Meet Claim 15 for *Receiving an Edit Mode Entry Signal ... From the Client Terminal Representing Entry into the Cue Point Edit Mode, and Switching the Video Editing Environment to the Cue Point Editing Mode in Response to Receiving the Edit Mode Entry Signal*

279. Claim 15 recites that an *edit mode entry signal* is sent “*over the network*” from the client to the server-side that is separate from the *cue point editing*

command of limitation 13.4 is that its sent over the network. The “*VEE switch[es]* ... *to the cue point editing mode in response to receiving the edit mode entry signal.*” Thus, Claim 15 calls for two separate signals sent over the network from the client to the server side: (1) an *edit mode entry signal* and (2) the *cue point editing command*.

280. This understanding of the *edit mode entry signal* is supported by the claim language and specification of the ’286 Patent, as discussed above regarding Gupta in Section VIII.F.

281. In its Institution Decision in the -00710 IPR, the Board raised a question as to whether Kim teaches Claim 15. See Ex. 2034 (Decision on Institution, Paper 13, 9/5/2024), 36 (calling into question “whether Kim discloses Limitations 13.1, 13.4, 13.5, and Claim 15”).

282. The Petition and its expert assert that the claimed *cue point editing mode* is entered in a first case when an administrator at client 230 selects the “Save” button 1822 in the view advertisement window of Figure 18A, which leads to the view advertisement window of Figure 18B, which allows an administrator to “edit video title or allowable playing duration.” Pet., 66. Kim, [0334]-[0335], Figs. 18A-18B. The Petitioner asserts that the claimed *cue point editing mode* is entered in a second case when a client 230 selects the “add-bookmark-title control button 922” in Figure 9 of Sull. Pet., 67 (citing Ex. 1007 (Sull), [0213]). See Sull, Fig. 9 (control

button 922). See Houh Decl., ¶ 274. In paragraph [0213], Sull states: “FIG. 14 is a flow chart illustrating the process of adding a title to a multimedia bookmark.”

283. The Petitioner thus argues that the *edit mode entry signal* transmitted by the client “*over the network*” to the server-side is met by (1) Kim’s user selecting the Save button 1822 in Figure 18A or (2) Sull’s user selecting the add-bookmark-title control button 922 in Figure 9 as referenced in Figure 14. Pet., 68. See Sull, Figs. 9, 14. Houh Decl., ¶¶ 274-275.

284. Kim discloses the following regarding the Save button 1822 in Figure 18A: “The administrator then completes the selection [of advertising bookmarks] by clicking on save button 1822, and a new GUI screen 1830 of FIG. 18B will appear.” Kim, [0334], Figs. 18A-18B. Thus, Kim discloses that selecting save button 1822 in Figure 18A causes a local operation of the client transitioning to the view bookmarks window of Figure 18B. There is no teaching or suggestion that the save button 1822 triggers the transmission of a command signal (*edit mode entry signal*) from the client 230 over the network to the server-side per Claim 15.

285. The Petitioner also argues that selecting Sull’s add-bookmark-title control button 922 is the claimed *edit mode entry signal*. Pet., 68. For the same reasons explained above in Section XI.C.1, the user’s selection of the add-bookmark-title control button 922 does not trigger the client to transmit a command signal “*over the network*.” Nowhere does Sull teach or suggest that the user selecting

the add-bookmark-title control button 922 causes the client to transmit an *edit mode entry signal* “over the network” to the server-side. Instead, like Kim, it simply transitions the client software to the next step in the process (e.g., to “Title Input 1404?” in Figure 14 of Sull).

286. None of the Petitioner’s and its expert’s citations to Kim and Sull support the assertion that selecting the save button 1822 in Kim or selecting the add-bookmark-title control button 922 causes the client to transmit an *edit mode entry signal over the network* to the server-side. See Houh Decl., ¶¶ 266-278. Rather, the passages reflect Kim’s and Sull’s teaching that the client-side software (web browser 230, multimedia bookmark client 238) **locally creates and edits** bookmarks which are stored locally in local storage 236. Kim, Fig. 2. See Section XI.B. A POSITA would understand that the user interfaces for a user to create and edit bookmarks are locally-generated by the client-side browser 230 working with the client-side multimedia bookmark client 238. A POSITA would understand that the browser 230 can **retrieve** bookmarks from the multimedia bookmark BBS server 212 using http requests to web server 216. See Kim, Figs. 2-3, [0290]). However, the **editing** of bookmarks is carried out locally by the client software on client 230, as explained in Section XI.B. Accordingly, the client 230 does not transmit a command (*edit mode entry signal*) to the server-side to initiate the editing mode (*cue point editing mode*) as required by Claim 15.

287. The Petitioner’s expert argues that the “change from one interface to another would be based on a request (e.g., HTTP request) from client 230 to the web server.” Houh Decl., ¶ 275. This statement should be given no weight because it is not supported by citation to evidence. Additionally, there is no evidence that the presentation of Figure 18B after the user selects save 1822 in Figure 18A is caused by the client sending an additional HTTP request to the web server. A POSITA would understand that the user interface of Figure 18B can already be resident in the multimedia bookmark client 238, which may be “embedded in [the] Internet browser.” Kim, [0297], Fig. 1 (client 230). There is no evidence that selecting save 1822 causes the client to send an “*edit mode entry signal*” “*over the network,*” as claimed.

288. Accordingly, Kim and Sull do not teach or suggest Claim 15.

E. Kim Fails to Meet Claim 17 for *Generating, Automatically, By Using the Computer System ... A Portion of the Set of Cue Point Information*

289. The Petitioner asserts that Claim 17 is obvious based on Kim/Sull. Pet. (IPR2024-01101), 69-70. Claim 17 is patentable for at least the same reasons set forth for independent Claim 13.

F. Kim Fails to Meet Claim 18 for *Receiving, Over the Network, Authentication Information Related to Access Authority of the Remote Client Terminal*

290. In Claim 18, the remote client terminal has authentication information relating to access authority that is transmitted “*over the network*” to the server-side.

291. The Petitioner points to Kim’s disclosure of retrieving a user’s license key in order to access DRM-packaged video content. Pet., 69. See Kim, [0343]-[0345], Figs. 20A-20B. The license key is “used to unpack the packaged video.” Kim, [0344], [0345] (“The multimedia bookmark server then unpacks the requested DRM packaged video stored at storage 2022 with the returned license key” and then provides an “extracted multimedia bookmark image” to the client). Accordingly, the “information on the user” and “returned license key” are used to allow the client to receive video (bookmark image), **not to perform editing** of bookmark information as required by Claim 18 in conjunction with limitations 13.4-13.5.

292. Accordingly, I disagree with the Petitioner and its expert regarding Claim 18. Pet., 68-69; Houh Decl., ¶¶ 279-281. Claim 18 is patentable over Kim.

G. Claims 14-15 and 17-20 Are Patentable Over Kim

293. Claims 14-15 and 17-20 are patentable over Kim for at least the same reasons set forth for Claim 13.

XII. CLAIMS 13-15 AND 17-20 ARE PATENTABLE OVER KIM AND GUPTA (GROUND 4)

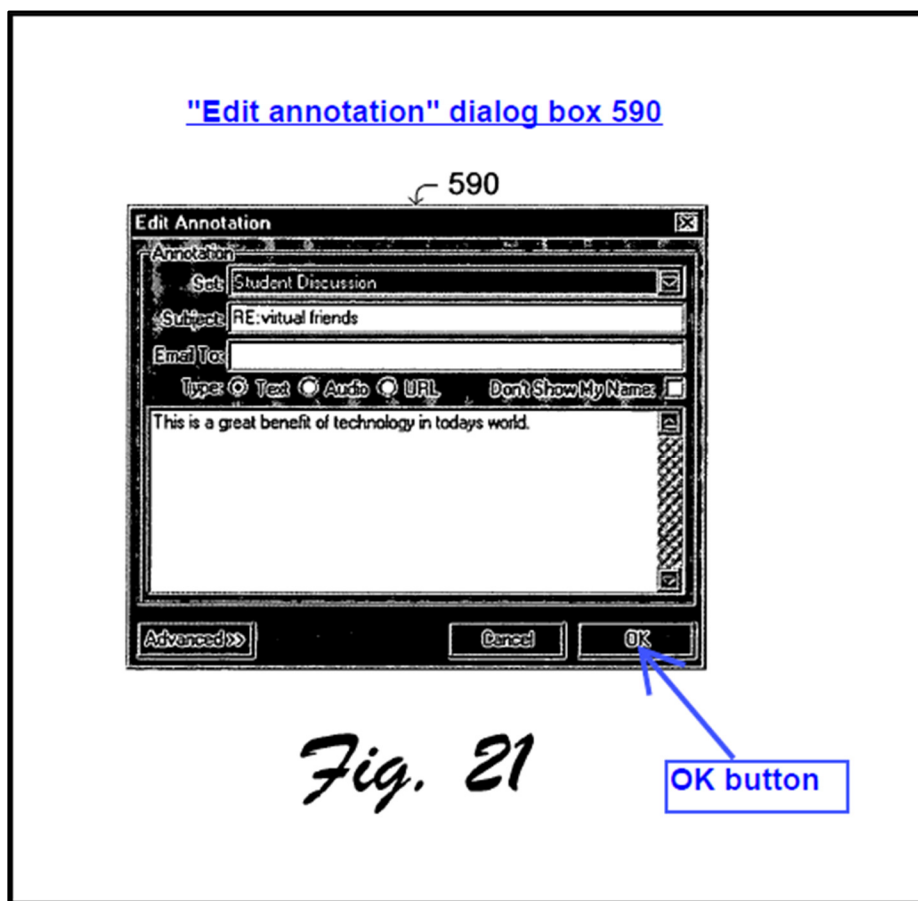
A. Independent Claim 13 Is Patentable Over Kim and Gupta

294. The Petition applies Gupta to Kim only to address limitation 13.4 and limitation 13.5 for receiving a “*cue point editing command*” “*over the network*” from

a remote client terminal and “*modifying a portion of the cue point information at the server side ... according to the cue point editing command received ... from the client terminal.*” Pet., 71-72. The Petition does not apply Gupta to address any other limitations from Claim 13.

295. The Petitioner relies entirely on Kim for all other limitations of Claim 13. Pet., 71. Accordingly, Claim 13 is patentable over Kim and Gupta because Kim does not teach or suggest limitation 13.1 for the reasons set forth above for Ground 3 (Kim) in Section XI.B.

296. For limitation 13.4, the Petitions relies on Gupta’s disclosure of the “OK” button in the Edit annotation dialog box 590 of Figure 21 that causes new annotation information to be sent to the server. Gupta, Fig. 21, [0100]. Pet., 71-72. See Section VIII.D.1.b above (explaining Edit annotation dialog box of Gupta, Fig. 21).



Gupta, Fig. 21 (“Edit annotation” dialog box 590). See Pet., 71-72 (citing Gupta, [0100], OK button “sends the new annotation information to the server”).

297. Regarding the question of motivation to combine, the Petitioner and its expert argue for combining the teaching of the OK button in Gupta with Kim because it would provide the “client-side editing functionality disclosed by Gupta” that “would have been appreciated” by a POSITA. Pet., 72; Houh Decl., ¶¶ 290-293. However, Kim already provides client-side editing functionality based on the client-side multimedia bookmark client 238 in client 230 which locally creates and edits bookmarks that are locally stored in local bookmark storage 236, as discussed in

detail in Section XI.B.1. Accordingly, a POSITA would not have reason to combine Gupta with Kim to provide client-side editing functionality because Kim already has that attribute.

298. Accordingly, Kim and Gupta does not meet Limitation 13.4 of Claim 13.

299. As for limitation 13.5 (*modifying a portion of the cue point information ... according to the cue point editing command*), the Petition provides no analysis based on the combination of Kim and Gupta. Pet., 71-72. The Petitioner relies on its analysis of Kim in Ground 3 for this limitation. However, Kim does not teach or suggest limitation 13.5 for the reasons set forth above for Kim for Ground 3 in Section XI.C.2 above. Gupta does not teach this limitation for the reasons set forth in Section VIII.E above.

300. Accordingly, Kim and Gupta does not meet limitation 13.5 of Claim 13.

301. Claim 13 is nonobvious over Kim and Gupta because the combination fails to meet multiple limitations of the claim.

B. Kim and Gupta Fail to Teach or Suggest Claim 15 and Claim 18

302. For Claim 15, the Petitioner does not provide a separate analysis based on Kim and Gupta, instead relying on its analysis for Kim in Ground 3. Pet., 72.

Accordingly, Claims 15 and 18 are patentable over Kim and Gupta for at least the same reasons set forth in Sections XI.D and XI.F above for Ground 3.

C. Claims 14-15 and 17-20 Are Patentable Over Kim and Gupta

303. Claims 14-15 and 17-20 are patentable over Kim and Gupta for at least the same reasons set forth for Claim 13.

XIII. CLAIMS 13-15 AND 17-20 ARE PATENTABLE OVER VASILEVSKY AND LOGAN (GROUND 5)

A. Claims 13-15 and 17-20 Are Patentable Over Vasilevsky and Logan

304. The combination of Vasilevsky and Logan fails to teach or suggest at least the following limitations of Claim 13:

- “*providing a computer system having at least one computer readable medium comprising a video editing environment, and being disposed at a server side of a public network*” (Limitation 13.1);
- “*receiving a set of video information at the video editing environment using the computer system*” (Limitation 13.2);
- “*receiving a set of cue point information at the video editing environment using the computer system*” (Limitation 13.3); and
- “*receiving a cue point editing command over the network from a remote client terminal*” and “*modifying a portion of the cue point*”

information at the server side of the public network according to the cue point editing command” (Limitations 13.4-13.5).

B. Vasilevsky and Logan Fail to Meet Limitation 13.1 for *Providing a Computer System ... Comprising a Video Editing Environment, and Being Disposed at a Server Side of a Public Network*

305. As discussed above in Section V.C and VIII.C.1, during the prosecution of the ’286 Patent the inventor repeatedly explained the server-side *video editing environment* application(s) for performing editing of cue point information that enables **remote cue point editing** by the client terminal as opposed to local cue point editing using a local client application. This understanding is supported by the ’286 Patent specification, also discussed in Section VIII.C.3. Limitation 13.1 reflects this requirement, reciting “*providing a computer system ... comprising a video editing environment, and being disposed at a server side of a public network.*”

1. The Assertion that Vasilevsky’s Home Area Network Is a Public Network Is Not Credible

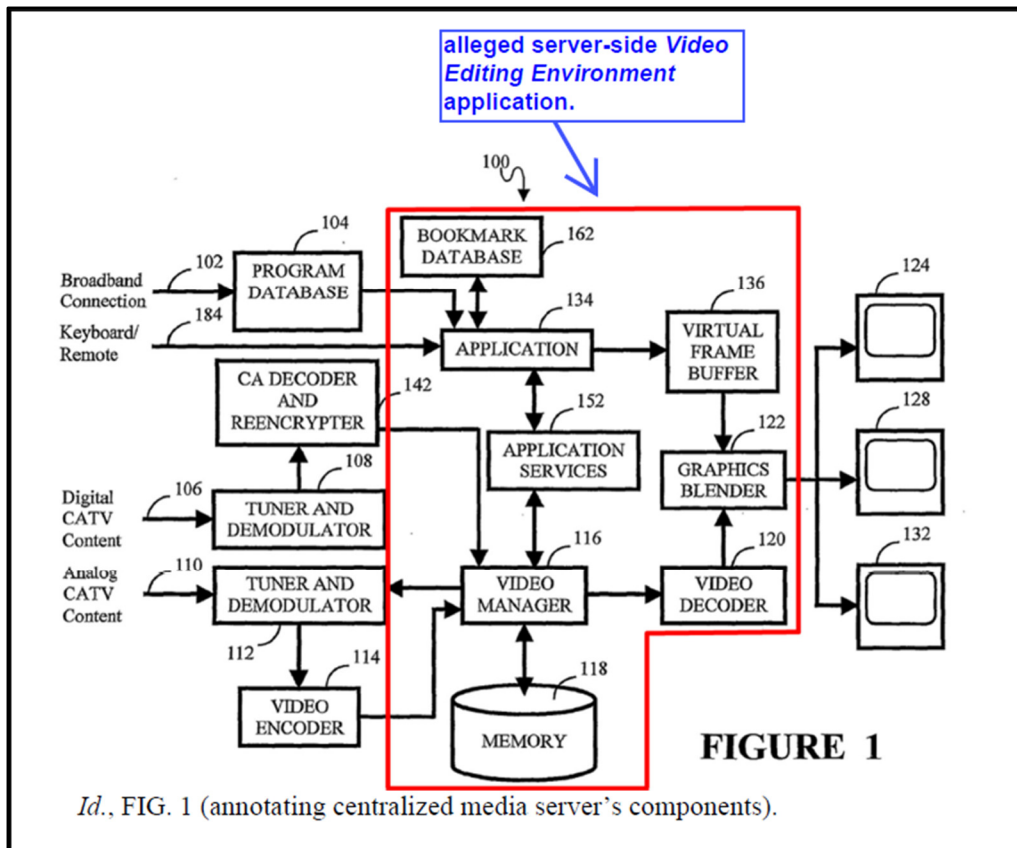
306. Petitioner’s expert asserts that Vasilevsky teaches that the disclosed Home Area Network is a public network. Houh Decl, ¶320 (“Vasilevsky describes such computer system being disposed at the server side of a public network (*e.g.*, [home area network] HAN ...”). I disagree with this assertion. Vasilevsky teaches that the Home Area Network is a private network, not a public network. The Vasilevsky reference also describes the HAN as “essentially a computer bus” in the home. This is further evidence that it is not a public network. Vasilevsky, 11:9-13.

Also, it is well-understood in the art that Home Area Networks are private networks, not public networks. Ex. 2039 (Alcatel), 2040 (Parolari).

2. Vasilevsky and Logan Do Not Meet the Claimed *Video Editing Environment* Application Because Vasilevsky Does Not Disclose Any Software Application for Editing Bookmarks

307. The Petitioner asserts that the claimed “*computer system ... comprising a video editing environment ... disposed at the server side*” of Limitation 13.1 is met by selected components of the “centralized media server” of Vasilevsky shown in Figure 1. Specifically, the Petitioner and its expert contend that the claimed *video editing environment* is the combination of Video Decoder 120, Graphics Blender 122, Application Unit 134, Virtual Frame Buffer 136, Application Services Unit 152, and Video Manager 116 and storages Memory 118 and Bookmark Database 162).” See Pet., 74; Houh Decl., ¶ 319.

308. The Petitioner’s alleged *computer system ... comprising a video editing environment* is illustrated in the Petitioner’s red-box mark-up of Vasilevsky:



Pet., 79 (Fig. 1, Vasilevsky, annotated by Petitioner in red; PO annotation in blue); Houh Decl., ¶ 319.

309. I have reviewed Vasilevsky in detail, and it does not disclose any specific software application for editing bookmarks on any component in the alleged *video editing environment* of Figure 1 for editing bookmarks.

310. Vasilevsky discloses the following pertaining to the processing of bookmarks (all *emph. added*):

The present home area-networked digital video recording and playback system 100, **capable of program bookmarking** and playback resumption of the same program from multiple receivers, is

explained below with reference to Figure 1. In its elementary form, the system 100 combines a **media server** for receiving and storing multiple electronic audio-visual programs (e.g., digital or analog television broadcasts, video recordings such as those provided by video-on-demand services, or even audio programs), and several reproduction devices such as digital or analog television sets (124, 128 and 132) via a home area network (HAN). The media server acts as a centralized Digital Video Recorder and Playback device for all of the receivers connected to the HAN.

Vasilevsky, 10:11-11:2.

The system users communicate with the media server via a user interface or **Application Unit 134, which converts user commands to commands compatible with the media server**. The Application Unit 134 is also responsible for generating a graphical user interface (GUI) containing an electronic program guide (EPG) for display, and [sic] input by an Application Services Unit 152. The Application Services Unit 152 controls the recording and playback of programs. Under the direction of the Application Unit 134 and utilizing program guide information, the Application Services Unit 152 establishes the necessary memory needed for recording a particular program, whether the recording is of the traditional or live-pause variety.

Vasilevsky, 15:15-16:5.

In operation, a viewer can retrieve programming from the server from any of the networked receivers. The same program can be watched simultaneously by more than one receiver if desired. Further, live-pause

control and playback can be executed from any networked receiver, allowing a program to be viewed in the live-pause mode in a seamless manner from more than one receiver. For example, a program being viewed on one receiver can be paused. Later, the same program can be resumed from the same pause point and viewed, but from a different receiver.

Therefore, both the viewing and control of programming can be distributed among multiple receivers in a seamless manner if desired. It should be appreciated by those skilled in the art that all trick play modes can be distributed across multiple receivers for a single program as described above with respect to the “pause” mode. For example, a viewer who has previously paused or rewound a program being recorded in live-pause mode can decide to “fast forward” to another point using one receiver, and then resume viewing from the stopping point (of the fast forward operation) using another receiver.

Vasilevsky, 18:13-19:11.

When a system user pauses the playback of a particular program, he or she can decide whether to “bookmark” the program. That is, an electronic marker can be placed in memory indicating [1] exactly where the program was paused, along with other particulars such as [2] the identity of the user, and [3] the date and time of the bookmark. When a command is received to enter a bookmark, the Application Unit 134 causes the bookmark to be stored in a bookmark database 162, along with [4] program identity information so that the bookmark can be matched to the appropriate program stored in memory 118.

Vasilevsky, 19:12-20 (brackets [1]-[4] added).

In the preferred embodiment, any **user with access to the system can place or establish program bookmarks in a program** being reproduced. **Any user can also access the bookmarks of other users and resume the playback of any available program from any bookmark**, regardless of who established it. Furthermore, **bookmarking and resumption of playback can be carried out from any system receiver.**

Vasilevsky, 20:1-10.

Figure 2 illustrates an example of a program guide display 200 presented to a system user, which display contains previously stored programs on the media server, or those in the process of being viewed in the live-pause mode. By highlighting a particular program, more detailed information can be obtained from the “Program Details” display 300 in Figure 3. The “Program Details” display 300 allows a user to reproduce the particular program from the start, **or from a previously established bookmark.**

Activating a “bookmarks” button in the display [sic: 300] causes the GUI to produce a toolbar or the like, which shows the bookmarks that have been established for the program. The **bookmark display 400** [of Figure 4] contains a start time display area 410 and end time display area 420, indicating the program length. A program length bar 430 is shown in the middle of the display. In the example shown, **a bookmark icon 440 represents a program bookmark that has been established by User Schmidt (identifier 450).** Position 460 can represent a **previous bookmark of a user** that has been updated to a

new position. The **bookmark display 400 can contain all of the active bookmarks** for the program, or only those to which the user has access. Vasilevsky, 21:1-19, Figs. 2-4.

Figure 6 shows a variation 600 of the bookmark display 400 in Figure 4. The elements 610, 620, 630, 640, 650 and 660 are analogous to the elements 410, 420, 430, 440, 450 and 460, respectively. However, the identifier 650 is in the form of a more detailed text box [of the bookmark] that includes such information as the bookmark establisher's name, a title of the bookmark, and a brief description of what occurred in the program just prior to the pause point. **A system user can enter or modify all of the information.** Vasilevsky, 22:8-15, Fig. 6.

Another variation of showing bookmarks isn't iconic at all, but rather completely textual, as shown in the display 700 of Figure 7. In that figure it can be seen that **all of the bookmarks accessible to system users are listed by program, user and program pause location.** Using this approach, it is also possible to gain more information about the bookmark by clicking a pointing device on the desired bookmark in the "program" column. Alternatively, or in conjunction, **clicking on the bookmark can lead directly to the pause point in the program of the representative bookmark.** A user can see other bookmarks established by other system users by clicking on the other users' names in the "user" column. In the example shown, two different system users (Publius Virgil and Mary Shelley) have placed bookmarks in the same program ("Home Improvement"). Vasilevsky, 22:16-23:8.

For example, the description of program bookmarks has been directed to **allowing the user to decide when to bookmark a program**. It is also possible for the system to **automatically establish a single program bookmark** when a program being reproduced is paused. Using this approach, the system might also automatically overwrite an existing bookmark with a new bookmark when the reproduction pauses at the same point as that represented by the old bookmark

Vasilevsky, 23:14-20.

311. Figure 6 illustrating the Bookmark display including the detailed textbox 650 is presented below:

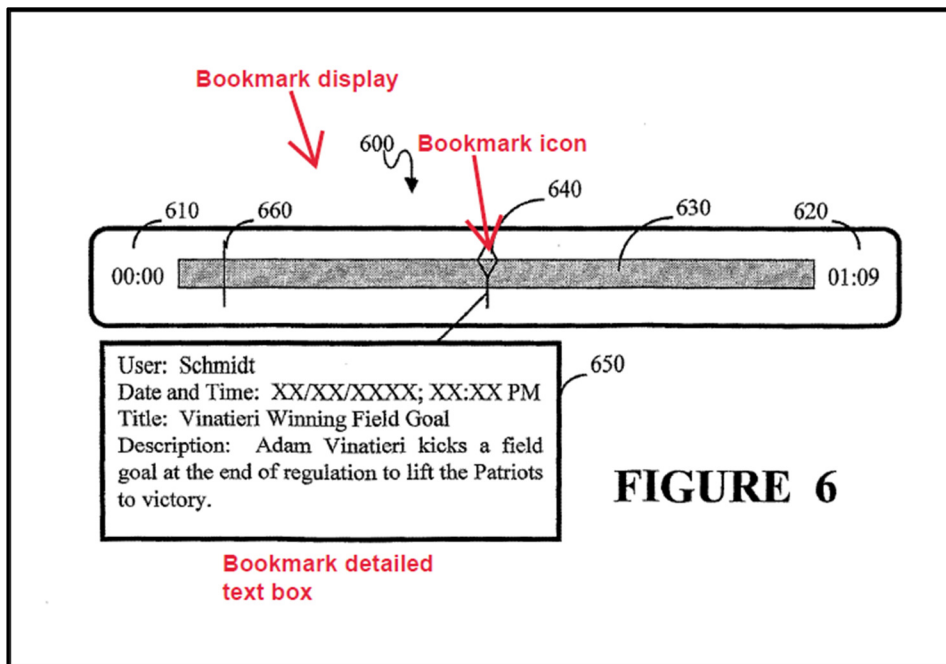


FIGURE 6

Vasilevsky, Fig. 6 (annotated).

312. The Petitioner and its expert Dr. Houh rely completely on Vasilevsky³ for meeting the server-side *video editing environment* application of limitation 13.1. Pet., 74-78; Houh Decl., ¶¶ 315-326. As can be seen from the passages cited above in this section, Vasilevsky does not describe that any of the components in the alleged *VEE* executes a software application for editing bookmarks. For that reason, the combination with Vasilevsky fails to meet limitation 13.1.

313. Additionally, the Petitioner and its expert have presented no argument that it is inherent that any of the components in the Petitioner’s red-boxed alleged *video editing environment* executes a software application for editing bookmarks per limitation 13.1. See Pet., 74-78; Houh Decl., ¶¶ 315-326.

³ The Petitioner and its expert rely on Logan only for the “*disposed at a server side*” aspect of Limitation 13.1, i.e., that a POSITA would rehost the alleged *video editing environment* of Vasilevsky onto the public Internet in place of the home area network (HAN). Pet., 76 (“To the extent Vasilevsky does not explicitly disclose its network (HAN) is a “public” network, Logan discloses this aspect.”); Houh Decl., ¶ 321 (same).

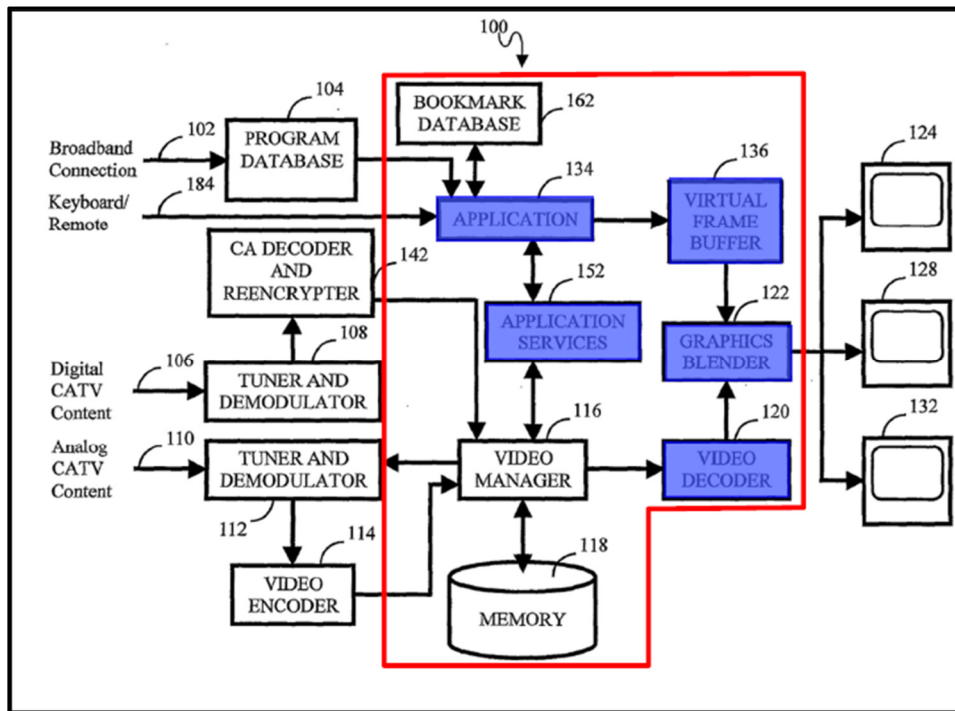
3. The Petitioner's Citations to Vasilevsky Fail to Teach the Claimed Server-Side *Video Editing Environment* Application

314. None of the Petitioner's and its expert's citations to Vasilevsky demonstrate a teaching of the claimed *video editing environment* application for editing bookmarks (cue points). For example, the Petitioner and its expert cite Vasilevsky at 20:11-17, which states:

While the components 120, 122, 134, 136 and 152 are part of the centralized media server in the preferred embodiment, it should be understood that **these components** might be **implemented in a client system or other end-user equipment** (e.g., 124, 128 and 132), in an alternate embodiment. While today's processor and memory prices dictate that **the aforementioned components** are better incorporated **in the centralized media server** to serve "thin" clients, future price reductions (and/or performance increases) in these elements may make the inclusion of these components in client or end-user equipment economically feasible.

Vasilevsky, 20:11-17 (emph. added). See Pet., 74-75; Ex. 1002 (Houh Decl.), ¶¶ 318-319 (citing same).

315. A POSITA would understand this passage as indicating that the cited "those components" (video decoder 120, graphics blender 122, application unit 134, virtual frame buffer 136, and application services unit 152) could be removed from the centralized media server (as illustrated below in **blue**) and could instead be incorporated into a client device (e.g., receivers 124-132), as illustrated below:



Vasilevsky, Fig. 1 (red annotation by Petitioner as alleged *video editing environment*; blue annotation indicating components removed and integrated into receivers 124-132 according to alternative embodiment at 20:11-17).

316. Specifically, the statement that “these components [video decoder 120, graphics blender 122, application unit 134, virtual frame buffer 136, and application services unit 152] might be implemented in a client system or other end-user equipment” (Vasilevsky, 20:11-15) just indicates those components could be implemented in the end-user receivers/set-to-boxes. It does not disclose or remotely suggest anything about any particular **software application** for editing bookmarks being executed by any of these components, which are allegedly part of the alleged *video editing environment* per Limitation 13.1. The statement merely indicates that

the identified components could be moved from the central media server into receivers 124-132.

317. The Petitioner and its expert also assert the following regarding the claimed server-side *video editing environment* application:

The centralized media server **executes software** for editing bookmarks and streaming video. Id., Abstract (“centralized media server (152) allows system users (124, 128, 132)” to “establish[]” and “access” “bookmarks”), 19:16-18 (“Application Unit 134 causes the bookmark to be stored in bookmark database 162”), claim 5 (“**media server is adapted to enable** a particular program bookmark to be edited”), claim 11 (“wherein said **bookmark indicia are modifiable by a user**”), 7:13-17 (“The media server is further adapted to allow a user to ... **resume reproduction of the program from the program bookmark.**”).

Pet., 75 (orig. emph.) (yellow highlight added); Ex. 1002 (Houh Decl.), ¶ 319.

318. However, none of those passages disclose that any component in the alleged *VEE* runs any particular software application. They certainly do not teach a server-side application that performs editing of cue points/bookmarks as claimed. At most, they describe that application unit 134 purportedly has the functionality for creating bookmarks upon user input (Vasilevsky, 15:15-17, 19:16-20); that those bookmarks are stored in the bookmark database 162 (Vasilevsky, 19:16-20); that they can be used by a user to resume playback of a stored program (Vasilevsky, 7:13-17, 8:9-11, 18:13-19:2); and that those bookmarks can be edited by a user

(Vasilevsky 22:10-15, Fig. 6). These passages do not teach or suggest that any component in the Petitioner's alleged *video editing environment* is "execut[ing] software," as alleged (see Pet., 75), to provide a server-side *video editing environment* application for editing bookmarks. Likewise, claims 5 and 11 of Vasilevsky are describing functions, not a *video editing environment* software application.

319. I note further that Vasilevsky does not reference "software," "application," or "computer program" in connection with any component of the alleged *video editing environment* having video decoder 120, graphics blender 122, application unit 134, virtual frame buffer 136, application services unit 152, and video manager 116. See Pet., 74; Houh Decl., ¶ 319.

320. A POSITA would understand that each of the components Petitioner identifies in the alleged *VEE* could have its functionality implemented as hardware (e.g., hardware including gates for logical operations) as opposed to being implemented using software. A POSITA would thus understand that it is not inherent that any of the alleged components in the alleged *VEE* runs a software application for editing cue points/bookmarks. I note also that neither the Petitioner or Dr. Houh makes an inherency argument that the any of the components in the alleged *VEE* inherently runs a software application for editing bookmarks. See Houh Decl., ¶¶ 315-326.

321. Indeed, the component described in Vasilevsky for creating a bookmark (adding a bookmark) is application unit 134. Specifically, the application unit 134 is described as accepting a user input [from the “Keyboard/Remote”] to add a bookmark as follows:

The system users communicate with the media server via a user interface or Application Unit 134, which converts user commands to commands compatible with the media server.

Vasilevsky, 15:15-17.

When a command is received to enter a bookmark, the Application Unit 134 causes the bookmark to be stored in a bookmark database 162, along with program identity information so that the bookmark can be matched to the appropriate program stored in memory 118.

Vasilevsky, 19:16-20.

322. As can be seen from the above passages, there is no disclosure that the application unit 134 is running a *video editing environment* application or executing a particular software application to add a bookmark. There is no reference to software at all.

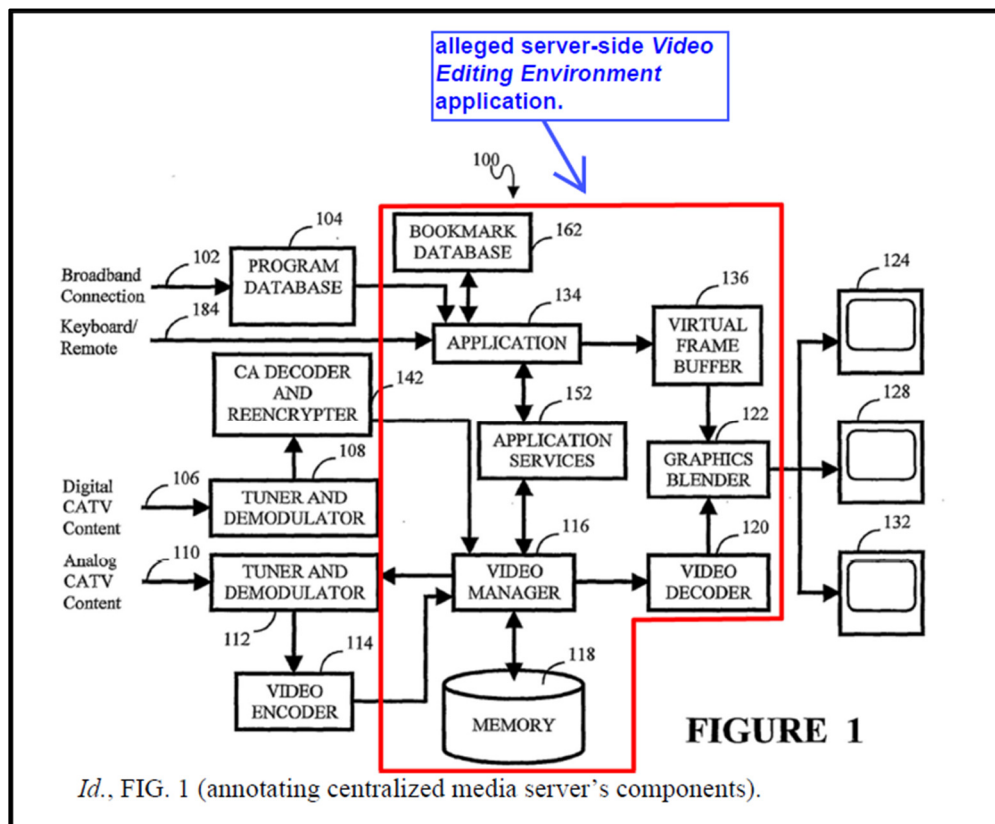
323. The Petitioner’s expert references Claim 1 of Vasilevsky, which recites “said media server is adapted to enable a particular program bookmark to be edited.” Houh Decl., ¶ 319. That passage does not teach or suggest that any component in the alleged *VEE* runs a software application for editing bookmarks, as Limitation 13.1 would require. In the same paragraph, Dr. Houh references Claim 7, which

provides that “said bookmark indicia are modifiable by a user.” Like Claim 1, Claim 7 does not teach or suggest that any component in the alleged *VEE* runs a software application for modifying a bookmark.

324. Therefore, it is my opinion that the combination of Vasilevsky and Logan fails to teach or suggest Limitation 13.1 for the *video editing environment*, which requires a software application for editing cue points/bookmarks.

4. A POSITA Would Not Modify Vasilevsky based on Logan to Rehost Vasilevsky onto the Public Internet

325. The Petitioner and its expert assert that the components in the red-box mark-up below correspond to the claimed “*computer system ... comprising a video editing environment ... disposed at a server side of a public network*”:

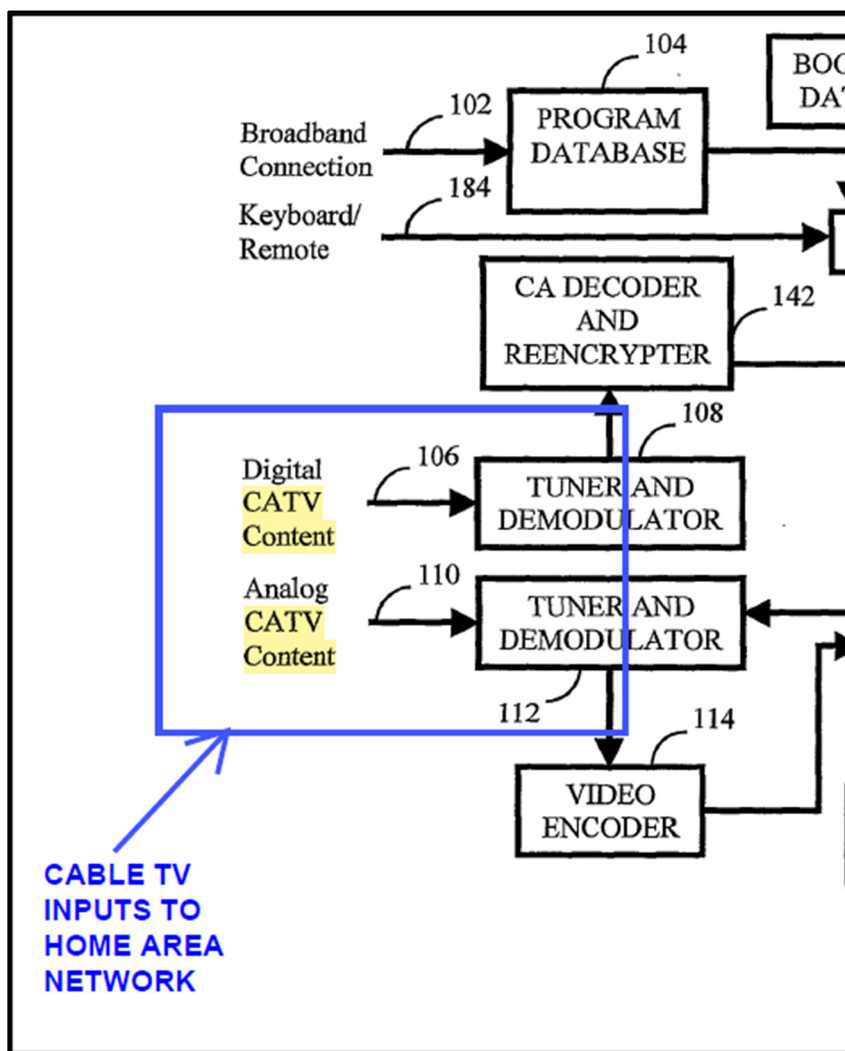


Pet., 75 (Fig. 1 of Vasilevsky annotated by Petitioner Aylo; annotated by PO in blue); Houh Decl., ¶ 319.

a. Modifying Vasilevsky to Be Hosted on the Public Internet in Place of the Home Area Network Would Not Work

326. The Petitioner and its expert argue that a POSITA could simply “modify Vasilevsky’s [HAN] network to be a public network,” that is, “swap[]” the public Internet in place of the HAN. Houh Decl., ¶¶ 322-325.

327. In my opinion, the proposed modification of Vasilevsky’s system would not work. For instance, Vasilevsky’s HAN receives the programs as analog and digital cable television transmissions over the cable lines. Vasilevsky, Fig. 1, 1: 16-17 (“Home Area Network Including Arrangement for Distributing Television Programming Over Local Cable”). “Via inputs 106 and 110 [connected to Digital CATV Content and Analog CATV Content, respectively], the system 100 also receives digital and analog television program signals to be processed by the tuner and demodulator circuits” in the home. Vasilevsky, 13:5-7. The connection of Vasilevsky’s home media server to the cable system enables the receipt of “digital or analog television broadcasts.” Vasilevsky, 10:14-19.



Vasilevsky, Fig. 1 (annotated in blue).

328. A POSITA would understand “CATV” in Figure 1 to refer to cable television transmissions delivered by a cable transmission facility such as a cable headend. A POSITA would understand that: (1) a cable headend has commercial grade front-end antennas and receivers to receive different program transmissions from multiple program content providers (e.g., satellite transmissions [e.g., HBO, ShowTime, ESPN, etc.], microwave transmissions, and over-the-air transmissions

[e.g., ABC, NBC, CBS, etc.]; (2) a cable headend receives and selects specific programs from the many incoming programs to repackage selected ones into a multiplexed program line-up transmission for the subscriber base; and (3) the cable headend transmits the repackaged multiplexed program line-up from the cable headend over cable (coaxial cable, hybrid fiber coax, or fiber) extending from the cable headend to individual homes of subscribers with their TV receivers. See Ex. 2017 (Hugenberg, Quest Communications), [0003], [0005], [0024]; Ex. 2016 (Wasilewski, Scientific Atlanta), 4:9-14.

329. If the Vasilevsky system is changed from the HAN network in the home to the public Internet, the system no longer has cable connections to the cable delivery system to receive the analog CATV program transmissions and digital CATV program transmissions. See Vasilevsky, Fig. 1 (cable connections 106, 110).

330. As proposed, the modified Vasilevsky system would no longer have cable connectivity and instead would purportedly have connectivity to the Internet. The modified system would not work because there would be no way for the modified system to receive analog and digital television programming transmissions over the cable infrastructure. See Houh Decl., ¶¶ 322-325.

331. The Petitioner and Dr. Houh do not explain how the Vasilevsky system would be able to receive analog and digital CATV program transmissions after “substituting” the HAN with the Internet, that is, rehosting the Vasilevsky system

onto the Internet. See Houh Decl., ¶¶ 322-325. Specifically, the Petitioner and its expert do not explain how the rehosted system would be able to receive cable transmissions to be stored in the DVR memory 118 and/or to be delivered directly to the end user receiver devices 124-132, as Vasilevsky requires. Vasilevsky, 13:5-15:4, Fig. 1.

332. Furthermore, the end user receiver devices 124-132 (which can be an analog television set or digital television set, see Vasilevsky, 10:18-19) would no longer be receiving analog and digital television content delivered via cable lines through the other components of Vasilevsky's home-based system. Instead, the end user devices would be client devices connected to the Internet. No explanation is provided by the Petitioner or Dr. Houh as to how, for example, an analog television set or digital television set connected to the Internet would be able to receive television programming from cable headends.

333. Dr. Houh relies on Ex. 1012 (Devore) to support his argument that the HAN can be "swapped for" or "substitut[ed]" for implementation onto the Internet, a change that would be allegedly be "invisible." See Houh Decl., ¶ 324. Devore describes a system connecting databases 110 with a first network 120 and a second network 160 to end users 150. Devore, [0032], Fig. 1. Devore states that the two networks "can be separate networks or alternatively, can be the same network and

can be, for instance, a private network or public network such as ... [the] internet.” Devore, [0032].

334. A POSITA would understand Devore to be teaching that the two networks can be part of the same network, or they can be different networks. Devore does not remotely teach or suggest that one network can be “swapped” for another network. In the case of a system implemented on a private network, Devore does not teach that the private network can be “swapped” or “simp[ly] ... substituted” with a public network like the Internet. Houh Decl., ¶ 324. With all due respect, Devore does not support the position taken by Dr. Houh at all. Accordingly, Dr. Houh’s argument that “it would have been a simple and straightforward modification of substituting ... a private network with a public network” is not supported by evidence. See Houh Decl., ¶ 324. There is only Dr. Houh’s conclusory reasoning, which does not withstand scrutiny for the reasons above where I explain why the proposed modification would not work.

335. Accordingly, the Petition’s obviousness theory does not establish a motivation to combine and reasonable expectation of success (see Pet. 77-78 (citing Houh Decl., ¶¶ 324-325)) because Devore does not show that a POSITA would understand that a private HAN network could be simply “swapped” or “substituted” with the public Internet.

336. Dr. Houh’s argument that Vasilevsky’s reference to a WAN provides a motivation to modify Vasilevsky to be reimplemented on the public Internet is not supportable. Houh Decl., ¶ 325. The cited section of Vasilevsky refers to the system accessing program guide data to populate the electronic program guide presented to end users. Vasilevsky, 12:5-10. A POSITA would understand that does not teach or suggest that the Vasilevsky system hosted on the Home Area Network be rehosted onto the Internet. Further, the Vasilevsky system is acting as a **client** receiving program guide data from a host server on the WAN, not as a **server side** system as claimed. Thus, the passage does not teach or suggest hosting the Vasilevsky system on the server side of a public network, as would be required by limitation 13.1.

337. Dr. Houh proceeds to argue that the modification would entail Vasilevsky’s system being implemented on a “single centralized server” on the Internet. Houh Decl., ¶ 326. However, neither the Petitioner or Dr. Houh explain how this “single centralized server” could receive cable television programming or how the end user devices could receive that programming, as I addressed above.

338. The proposed modified Vasilevsky system would not work and thus does not meet limitation 13.1.

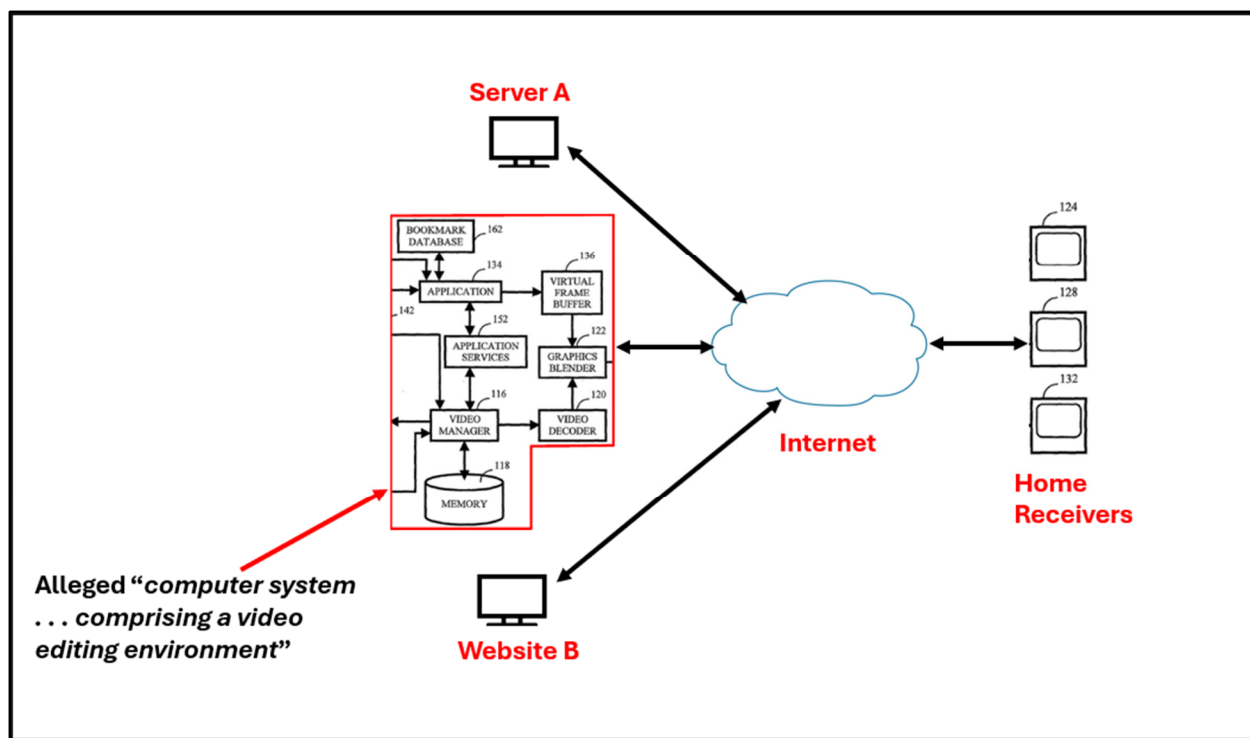
339. There is also no motivation to combine/modify and there is no reasonable likelihood of success for the reasons I set forth above. A POSITA would not modify Vasilevsky as proposed because the resulting system would not work.

For these additional reasons, the proposed combination does not meet limitation 13.1.

b. The Proposed Combination to Rehost the Alleged Computer ... Comprising a Video Editing Environment on the Internet Would Not Work Because There is No Front-End Receiver Circuitry to Enable Reception and Storage of Selected Programs as Required by Vasilevsky

340. As best can be understood, the Petitioner's proposed combination is based on rehosting the alleged *video editing environment* from Vasilevsky onto the public Internet. See Pet., 76-78; Houh Decl., ¶¶ 321-326. Dr. Houh states that “[a] POSITA would have been motivated to implement Vasilevsky's system on a public network [Internet].” Houh Decl., ¶ 323. Rehosting the alleged *video editing environment* onto the Internet produces a system like the following:⁴

⁴ This is not an improper bodily incorporation of references because it reflects the Petitioner's proposal: rehost the alleged *computer system ... comprising a VEE* (red box) from the Home Area Network (HAN) onto the Internet. See Pet., 77 (“modify Vasilevsky's network (HAN) to be a public network (Internet) ... simple and straightforward modification of substituting one network type with another”); Ex. 1002 (Houh Decl.), ¶ 324 (“simple and straightforward” matter of “swapping” one type of network for another).



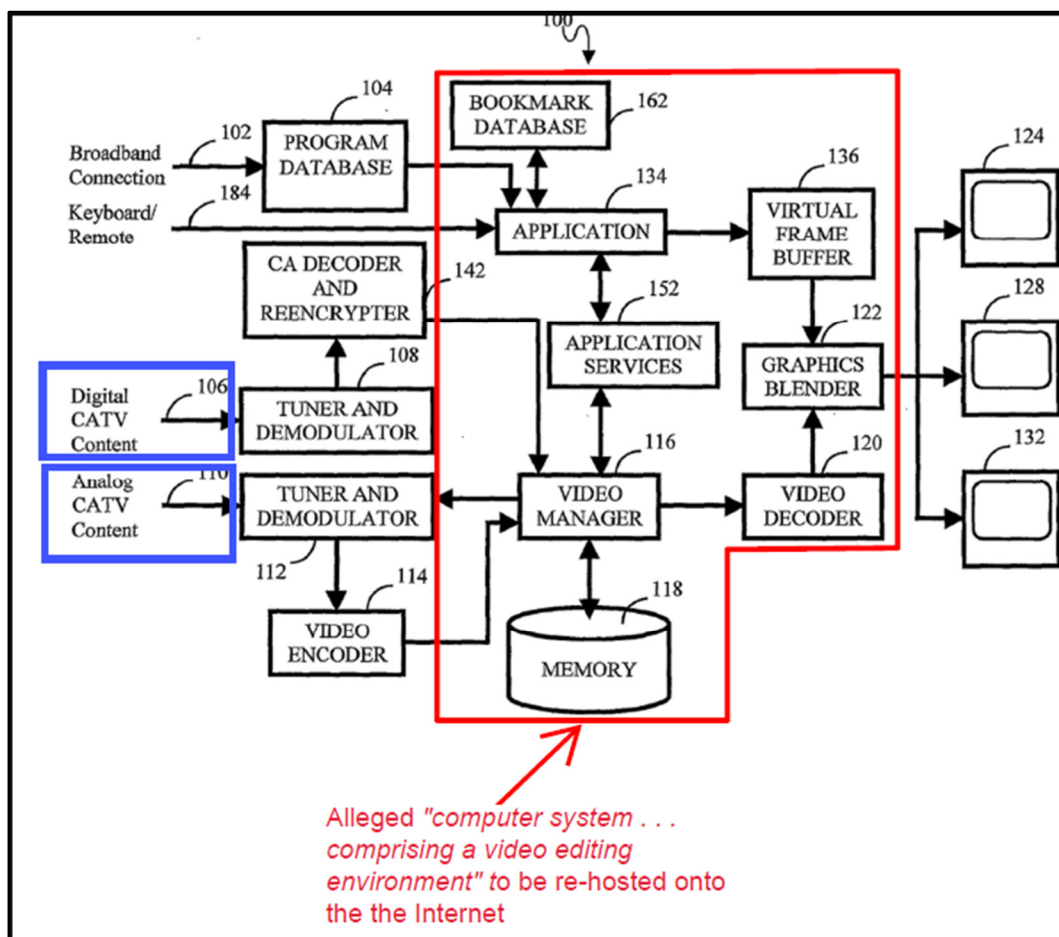
See Ex. 2022.

341. Therefore, a POSITA would readily understand that the Petitioner's and its expert's proposed combination has a significant problem because it results in a system where the rehoused components of the centralized media server (in Petitioner's red-box) would not be capable of:

- (1) receiving/demodulating analog television transmissions 110 to store program content in DVR memory 118 and/or to deliver it to end user devices 124-132;
- (2) receiving/demodulating digital television transmissions 106 to store program content in DVR memory 118 and/or deliver it to end user devices 124-132;

- (3) decoding/decrypting/re-encrypting the received digital television transmission prior to storage in DVR memory 118 and/or to deliver it to end user devices 124-132; and
- (4) encoding the received content from the television transmissions in the appropriate format prior to storage in DVR memory 118.

Vasilevsky, 13:5-15:4, Fig. 1.



Vasilevsky, Fig. 1 (blue annotation by Patent Owner; red-box indicated by Petitioner).

342. In other words, the Petitioner's proposed combination would render a system where alleged "*computer system ... comprising the video editing environment*" from Vasilevsky has no front end receiver interface components to tune, demodulate, decode, decrypt, re-encrypt, and encode the program transmissions to extract the desired program and provide it in a format suitable for storage in memory 118. See Vasilevsky, 10:15-16 ("receiving ... digital or analog program broadcasts"), 13:5-10 ("The tuner and demodulator circuits have the function of tuning into program channels selected to be received by the system 100 and then demodulating them in a manner known in the art."), 14:8-20 ("Prior to storage in memory, the analog program signals from the tuning and demodulating circuit 112 are digitized and then encoded via a video encoder 114, using a digital compression scheme such as those established by the Motion Picture Experts Group (MPEG). ... the digital program signals are decoded and re-encrypted by the circuit 142 prior to storage as an Elementary Stream in the memory 118."), 14:14-15:2 ("the digital program signals are decided and re-encrypted by the circuit 142 prior to storage as an Elementary Stream in the memory 118. . . . the circuite 142 can decrypt and then re-encrypt program streams"). See Vasilevsky, Fig. 1 (cable digital television transmission tuner/demod 108, cable analog television transmission tuner/demod 112, decoder and reenrypter 142, video encoder 114). See Ex. 2022.

343. A POSITA would understand that even if the system was implemented without the tuner/demodulators (108, 112), Vasilevsky requires the video encoder 114 to encode program signals into the appropriate format before they are stored in memory 118. Vasilevsky, 14:8-16, Fig. 1 (video encoder 114, memory 118). Similarly, a POSITA would understand that even if implemented without tuner/demodulators, Vasilevsky requires the decoder-reencryptor 142 to decrypt encrypted programming that is received and then re-encrypt it prior to storage. Vasilevsky, 14:14-15:2.

344. Accordingly, the proposed modification does not meet limitation 13.1 because it results in an inoperative system. Under that same reasoning, a POSITA would not have reason to make the proposed combination, and there would not be a reasonable likelihood of success that it would work for its intended purpose.

c. There is No Apparent Reason to Combine As Proposed and There Would Not Be a Reasonable Likelihood of Success Because of the Deficiencies in the Combination

345. For the reasons cited above in Section XIII.B.3, a POSITA would not have an apparent reason to make the proposed combination because Vasilevsky would not operate for its intended purpose of enabling users to select specific program units for receipt and storage into the shared DVR memory.

346. For similar reasons, the proposed combination would not have a reasonable expectation of success, which is a separate, fundamental requirement of

the obviousness case. In this case, there would not be a reasonable expectation of success because the proposed modified Vasilevsky system would not be able to perform its intended function of allowing users at receivers 124-132 to have their selected programs received and recorded onto the DVR memory 118. The Petition addresses motivation to combine but does not explicitly address a reasonable expectation of success. See Pet., 76-78. The arguments on motivation to combine assert that the combination would be a “simple and straightforward modification of substituting one network with another” (Pet., 77; Houh Decl., ¶ 324) but never address the details, specifically, how the red-boxed components (allegedly constituting the claimed *computer system ... comprising a video editing environment*) rehosted onto the Internet would be capable of receiving and delivering selected cable television programming for storage. Dr. Houh conclusory assertion “that such a substitution would not have affected the function of Vasilevsky’s system” (Houh Decl., ¶ 324) is given without any explanation or evidence. He fails to explain how the red-boxed components rehosted on the Internet could operate to receive and decode programming when they have no front-end circuitry to enable them to receive, decode, decrypt, re-encrypt, and encode programming before it is stored in memory 118, as discussed above. See Houh Decl., ¶¶ 324-326. Nor does he address how the re-hosted red-box components could receive digital and analog cable transmissions in his proposed modification.

347. For similar reasons set forth above, the proposed combination would fail because it would be inoperative for its intended purpose, and further, the combination would change the principle of operation of Vasilevsky, which is to enable multiple users in a home to select specific programs from incoming cable television transmissions to store them on a shared DVR memory.

348. Accordingly, the proposed combination of Vasilevsky and Logan fails to meet Limitation 13.1.

C. Vasilevsky and Logan Fail to Meet Limitation 13.2 for *Receiving a Set of Video Information At the Video Editing Environment*

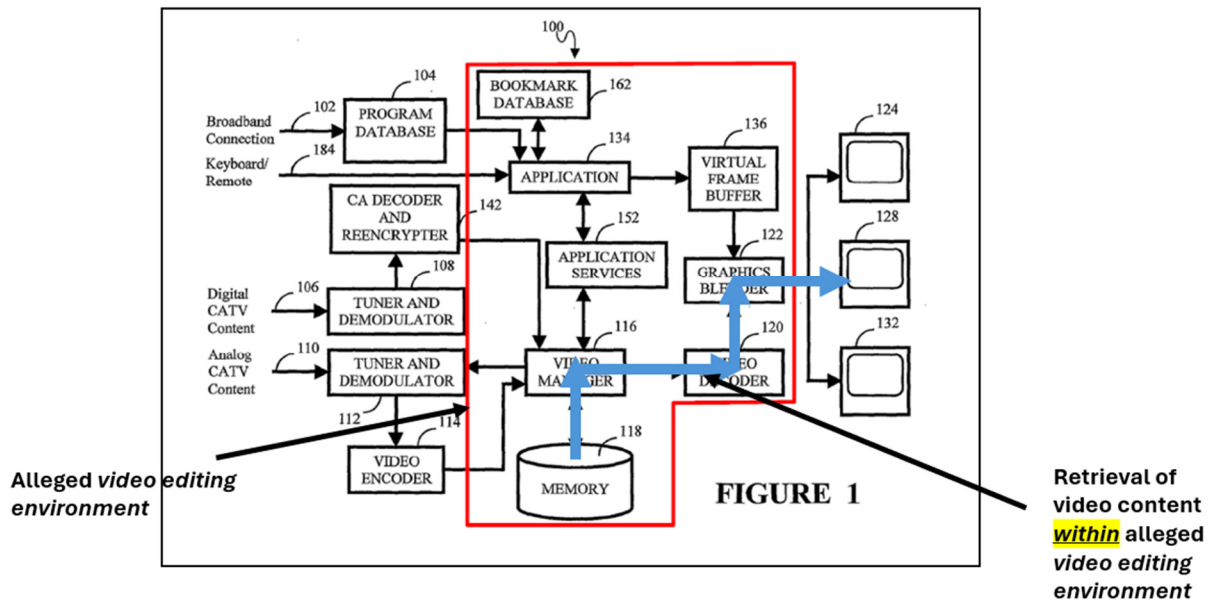
349. Vasilevsky and Logan does not teach “*receiving a set of video information*” at the VEE per limitation 13.2 so that cue point editing has occurred pursuant to limitations 13.4-13.5.

350. The Petitioner argues that the alleged red-boxed “*computer system ... comprising a video editing environment*” is receiving video information when retrieving it from memory 118 to stream it to client receivers 124-132. Pet., 78-79; Houh Decl., ¶ 329.

351. The Petitioner states: “The set of video information is received at the video editing environment using the computer system as video recordings are received by the software executed by Application Unit 134, Video Manager 116, and Memory 118, such as when retrieving video to stream to client 124.” Pet., 78-79 (citing Vasilevsky that “the reading of program content information from memory are controlled by a video manager 116.”) (orig. emph.); Houh Decl., ¶ 329.

352. However, those operations do not constitute receiving video information “at the video editing environment” per the claim. The memory 118 providing the video is *part of* the alleged *video editing environment*, so retrieving programming from memory 118 to deliver it to users it is an operation **internal** to the alleged “*computer system ... comprising the video editing environment*.”

353. The operation of the alleged red-boxed *video editing environment* retrieving video from memory 118 for delivery to one or more of the receivers 124-132 is illustrated below:



Vasilevsky, Fig. 1 (annotated in red by Petitioner; annotated in blue/black by Patent Owner).

354. As can be seen, there is no receiving of video information “*at the alleged video editing environment*” as required by Limitation 13.2. The retrieval of video is completely internal to the alleged *video editing environment*.

355. The Petitioner argues that the alleged *video editing environment* in Figure 1 of Vasilevsky would receive video from the analog and digital television sources (Digital CATV 106 into tuner/demod 108, Analog CATV 110 into tuner/demod 112). Pet., 78-79 (citing Vasilevsky, 13:19-20, “television signals and

other program signals are stored in memory”); Houh Decl., ¶ 328. Vasilevsky describes in unmodified Figure 1 that “[v]ia inputs 106 and 110, the system 100 also receives digital and analog television program signals ... The television signals and other program signals are stored in memory 118.” Vasilevsky, 13:5-20, Fig. 1.

356. However, the Petitioner and its expert are improperly analyzing the references individually as opposed to analyzing the combination they have proposed.

357. In the proposed modified configuration, as explained in Section XIII.B.4, there is no analog cable transmission or digital cable transmission received over the cable lines by the Vasilevsky system. Additionally, there is no front-end receiver circuitry to receive, tune, decode, and decrypt content from those (nonexistent) cable headends. Accordingly, a POSITA would understand that the alleged *video editing environment* is not receiving analog and digital television in the modified Vasilevsky system proposed by the Petitioner and its expert. Accordingly, the Petitioner’s proposed combination does not meet the requirement of “*receiving a set of video information at the video editing environment*” per limitation 13.2.

358. Even if the Vasilevsky/Logan combination allowed for the alleged *VEE* to receive a television transmission from a cable system, the *VEE* would not be receiving video information per limitation 13.2 so that a portion of it can be streamed back to the client terminal pursuant to limitation 13.7 after cue point editing has

When a system user pauses the playback of a particular program, he or she can decide whether to “bookmark” the program. That is, an electronic marker can be placed in memory indicating exactly [1] where the program was paused, along with other particulars such as [2] the identity of the user, and [3] the date and time of the bookmark. When a command is received to enter a bookmark, the Application Unit 134 causes the bookmark to be stored in a bookmark database 162, along with [4] program identity information so that the bookmark can be matched to the appropriate program stored in memory 118.

Vasilevsky, 19:12-20 (brackets added).

361. Accordingly, individual bookmarks are created by application unit 134 **within** the alleged *video editing environment* and are received (stored) at bookmark database 162 **within** the alleged *video editing environment*. This does not satisfy limitation 13.3 because the cue points (bookmarks) are not being received “**at the** *video editing environment*.”

362. In Vasilevsky, there is also a “bookmark-viewing” embodiment where a user at a receiver 124-132 viewing a program can view associated bookmarks retrieved from the bookmark database 162, and those bookmarks can be the user’s bookmarks or bookmarks created by other users in the home. Vasilevsky, 20:1-6. Figure 3 describes a “Programs Display” 300 including a “bookmarks” button that can be selected to provide a “Bookmark display” 400 as depicted in Figures 4-6. Vasilevsky, 21:4-19, 22:8-15. The Bookmark display includes a program time bar

and icons indicating bookmarks at various locations within the program. The user can view the content of bookmarks and modify bookmarks. Vasilevsky, Figs., 4, 6, 22:14-15 (“A system user can enter or modify all of the information” in the detailed text box for a bookmark).

363. A POSITA would understand Vasilevsky’s “bookmark-viewing” embodiment as describing that while a program is being delivered from memory 118, the corresponding bookmark information is delivered from bookmark database 162 to be received at the user’s receiver (124-132). In this instance, the alleged *video editing environment* is delivering, not receiving, bookmark information as would be required by Limitation 13.3.

364. The Petition argues that the alleged *video editing environment* receives a set of bookmark information in three ways. Pet., 79-80. The Petition first argues that “[t]he media server is further adapted to allow a user” of a client “reproduction device [to] place a program bookmark in a program representing a point in the program where reproduction has stopped.” EX1008, 7:14-16, 8:7-9. Pet, 80; Houh Decl., ¶ 332. This corresponds to the bookmark-creating scenario I discuss above.

365. I disagree with this assertion by the Petitioner and Dr. Houh for at least four reasons. First, the Petitioner’s argument describes the first scenario described above where a bookmark is created *within* the alleged *video editing environment* by application unit 134 and received by the bookmark database 162. These are

operations internal to the alleged *video editing environment*. Accordingly, this does not correspond to receiving bookmark information “**at** the *video editing environment*” as required by limitation 13.3.

366. Accordingly, this does not correspond to receiving bookmark information “**at** the *video editing environment*” as required by limitation 13.3.

367. The Petitioner makes a second argument that also corresponds to the “bookmark-creating” scenario described above. The Petitioner states that “[w]hen a system user pauses the playback of a particular program, he or she can decide whether to ‘bookmark’ the program” by placing “an electronic marker” “in memory” and “[w]hen a command is received to enter a bookmark, the Application Unit 134 causes the bookmark to be stored in a bookmark database 162.” *Id.*, 19:12-20, 22:8-15 (“A system user can enter or modify all of the [bookmark] information”). Pet., 80 (orig. emph.); Houh Decl., ¶ 333. For the reasons explained above, I disagree with the Petitioner and Dr. Houh because the creation of a bookmark within the alleged *video editing environment* in response to a user input is an **internal operation** in the *video editing environment*. The alleged *VEE* is not receiving bookmark information. The only receiving operation would be the bookmark database 162 receiving the bookmark after it is created by application unit 134, and that occurs **within** the alleged *VEE*. Thus, the referenced creation of a bookmark within the alleged *VEE*

does not meet Limitation 13.3 for “*receiving a set of cue point information at the video editing environment.*”

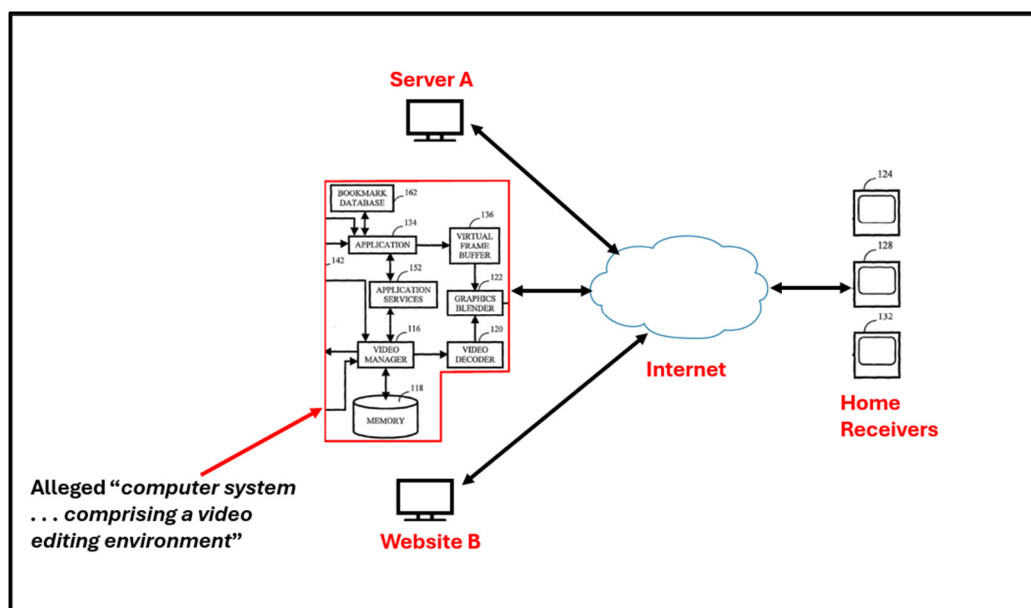
368. Similarly, the cited disclosure that a “system user can enter or modify all of the [bookmark] information” (Vasilevsky, 22:8-15, cited by Pet., 80 and Houh Decl., ¶ 333) is also referring to user input to create or modify a bookmark **within** the alleged *VEE*, as explained above. User input to create or modify a bookmark within the *VEE* does not meet limitation 13.3.

369. The Petition’s third argument is intertwined with its first two arguments. The Petitioner argues that “[t]he set of cue point information is received at the video editing environment using the computer system because [A] bookmarks are received by the software executed by Application Unit 134 and Bookmark Database 162, such as when receiving a new bookmark from client 124, or [B] retrieving a bookmark for client 124 to edit. Id., 15:15-20, 19:12-20; 21:1-8, 22:8-15; EX1002, ¶¶330-34.” Pet., 82 (brackets added).

370. As explained above, I disagree with the Petitioner and Dr. Houh because [A] the user’s entry of an input requesting the **creation** of a bookmark **within** the *VEE* does not meet Limitation 13.3, which requires the “*receiving of cue point information at the [VEE]*” for the reasons explained above. As for [B] “retrieving a bookmark for client 124 to edit” (Pet., 82), that refers to the alleged

VEE **transmitting** a bookmark to a receiver 124, not to the VEE **receiving** a set of bookmark information as would be required by limitation 13.3.

371. I also note that the proper focus of the inquiry is not on Vasilevsky alone. The correct focus is on whether the proposed combined system of Vasilevsky and Logan (*i.e.*, rehosting Vasilevsky onto the Internet) meets Limitation 13.3. In the proposed modified system, there is no mechanism for the Internet-hosted VEE to “receive a set of cue point information at the VEE,” as illustrated below:



See Ex. 2022.

E. Vasilevsky and Logan Fail to Meet Limitations 13.4 and 13.5 for Receiving a Cue Point Editing Command ... From a Remote Client and Modifying a Portion of the Cue Point Information at the Server Side ... According to the Cue Point Editing Command

372. Limitation 13.4 requires that the server-side “video editing environment” application receives a “cue point editing command” from the client

that, pursuant to limitation 13.5, causes the “modification of *the cue point information*” in response to the command. In other words, a *cue point editing command* causes the server-side VEE to edit “a portion of” the previously-received cue point information.

373. For limitation 13.4, the Petition asserts:

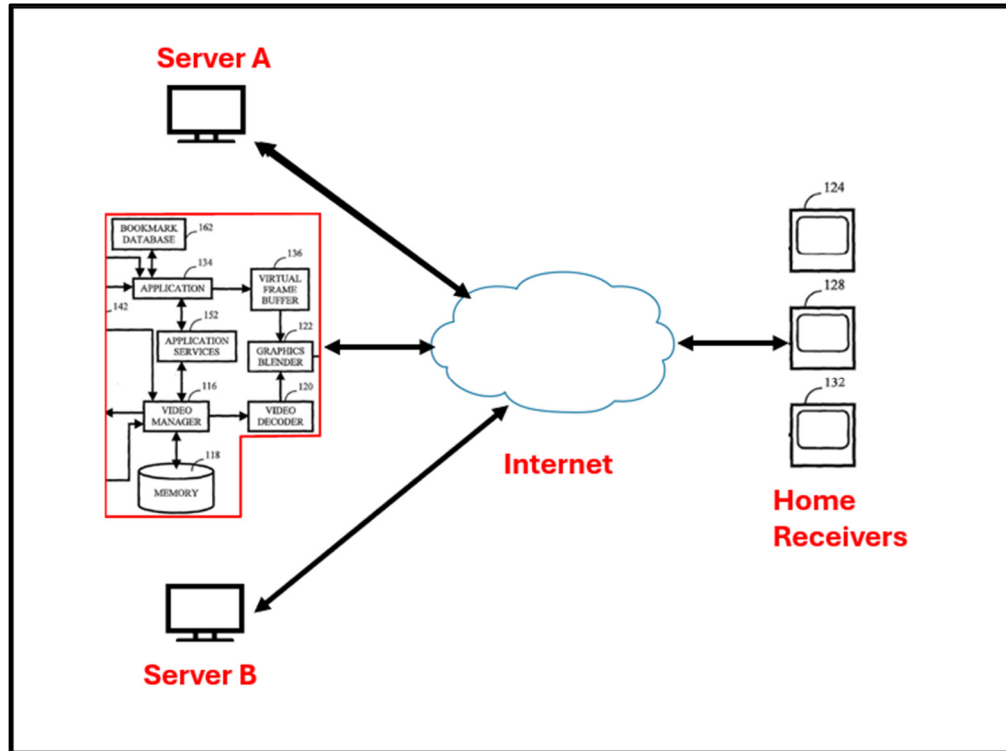
To edit a bookmark, client 124 enters an edit mode by selecting a bookmark, which displays the interface of Figure 6 showing a text box “includ[ing] such information as the bookmark establisher’s name, a title of the bookmark, and a brief description.” EX1008, 22:8-15. Client 124 can “modify all of the information,” **resulting in a command to edit the bookmark being sent from the client 124 and received using the centralized media server.**”

Pet., 83 (emph. added).

374. First, the Petitioner’s argument that a user’s entry of information to modify an existing bookmark is the *cue point editing command* for limitation 13.4 is without merit because the Petitioner argues that **the same user operation** meets “receiving a set of cue point information” for Limitation 13.3. Compare Pet., 82 (limitation 13.4, relying on Vasilevsky, 22:8-15) with Pet., 80-81 (limitation 13.3, relying on Vasilevsky, 22:8-15). Limitation 13.3 and limitation 13.4 are separately-recited limitations of the recited invention. The Petition cannot rely on the same operation in Vasilevsky to meet both limitations 13.3 and 13.4. For at least this

reason, the Petitioner has not established that Vasilevsky and Logan meet limitation 13.4.

375. Second, the Petition fails to address how limitation 13.4 is met by the **proposed combination** of Vasilevsky and Logan. I do not believe that it is sufficient to only point to disclosure in Vasilevsky. The Petition is improperly analyzing the individual references, not the references as combined. In the proposed combination, the alleged *VEE* is re-hosted on the Internet as follows:



See Ex. 2022 (modified, based on Pet., 79).

376. The Petitioner does not explain how, in this combined system, a home receiver 124 can issue a command over the Internet to be processed by application unit 134 to create a bookmark. See Pet., 83-84. If application unit 134 is to receive

an “input” over the public network from a receiver 124 located in the home, the application unit 134 would require some kind of interface (e.g., an IP address, a server-client interface, web server front end, etc.) to receive signals sent from the receiver over the Internet. However, the Petitioner and its expert do not address how application unit 134, hosted on the Internet, would be modified to be able to receive signals (e.g., an alleged *cue point editing command*) over the Internet. The Petitioner and its expert provide no prior art reference or citation to Vasilevsky to fill this gap in how limitation 13.4 is satisfied. See Pet., 83-84; Houh Decl., ¶¶ 336-346.

377. At the client, the Petitioner does not address how a local TV receiver 124 (variously described in Vasilevsky at 10:18-19, 12:3-4 as a “reproduction device,” “digital or analog television set[,]” and “television receiver”) would be modified to be capable of transmitting a command signal over the Internet to be processed by application unit 134. See Pet., 83-84. I believe that the TV receiver would need to be modified somehow to perform this operation, but the Petitioner and its expert do not address the necessary modification and provide no reference to fill this gap in the receiver 124 as disclosed. See Pet., 83-84; Houh Decl., ¶¶ 336-346. The proposed combined system does not meet limitations 13.4-13.5.

F. Vasilevsky and Logan Fail to Meet Claim 15 for *Receiving an Edit Mode Entry Signal Over the Network From the Client Terminal Representing Entry into the Cue Point Edit Mode, and Switching the Video Editing Environment to the Cue Point Editing Mode in Response to Receiving the Edit Mode Entry Signal*

378. Claim 15 in combination with Claim 13 provides that two different command signals are transmitted from the client over the network to the server-side: (1) an *edit mode entry signal* causing the VEE to switch ... to the *cue point editing mode*; and (2) a *cue point editing command* causing the VEE to edit the previously-received cue point information in accordance with the command. This is supported by the claim language and the specification of the '286 Patent in Section VIII.F regarding Gupta and Claim 15.

379. The Petitioner identifies an alleged signal from client 124 when the user elects to view/modify bookmark data in response to the Bookmark display GUI of Figure 6 of Vasilevsky. See Vasilevsky, 22:14-15, Fig. 6. Pet., 87. A POSITA would understand Figure 6 as indicating that a user can select an icon 640 to cause the bookmark details to be presented in bookmark details box 650. Vasilevsky states without any explanation that “[a] system user can enter or modify all of the information” in the bookmark. Vasilevsky, 22:14-15. Vasilevsky does not describe that any signal is issued when the user elects to “enter or modify” bookmark information. Vasilevsky, 22:8-15. The Petition vaguely refers to a “signal for entry to the interface(s) for editing bookmarks” without explanation or citation. Pet., 87; Houh Decl., ¶ 361. No such signal is discussed in Vasilevsky. Further, there is no showing by the Petitioner or its expert that when the user elects to enter/modify bookmark information from Figure 6 that a signal is transmitted “*over the network*”

to cause the alleged *video editing environment* to “switch ... to the *cue point editing mode*” in response to receiving the signal, as required by Claim 15.

380. Additionally, the Petitioner’s and its expert’s contention that the user electing to enter/modify bookmark information causes a signal to be transmitted over the network pursuant to Claim 15 is contradicted by their position for Claim 14 that the “*cue point editing mode*” is triggered by the user “selecting an existing bookmark” for editing. See Pet., 86-87; Houh Decl., ¶ 357.

381. The Petition (at 87) references Vasilevsky at 19:16-20, which states: “When a command is received to enter a bookmark, the Application Unit 134 causes the bookmark to be stored in a bookmark database 162.” There is no command transmitted “*over the network*” to cause transition into a “*cue point editing mode*” for **editing the bookmark** per Claim 15.

382. The Petition also references Vasilevsky at 22:8-15, which as explained above, describes that the user can access an existing bookmark and “can enter or modify” the information of the bookmark. Pet., 87. That passage does not disclose or suggest the receiver transmitting any particular signal, let alone an *edit mode entry signal* causing the VEE to switch into a *cue point editing mode*. Vasilevsky is completely silent on what operations the system performs to enable a user to edit a bookmark. See Vasilevsky, 22:8-15, Fig. 6.

383. The Petitioner states that “client 124 enters a cue point edit mode by **selecting an existing bookmark.**” Pet., 86; Houh Decl., ¶ 357 (emph. added). Thus, the Petitioner and its expert admit there is no *edit mode entry signal* because the alleged *cue point edit mode* is entered by the user “selecting an existing bookmark,” not transmitting a specialized *edit mode entry signal* over the network to Application Unit 134, as would be required by Claim 15.

384. Additionally, I find that the Petitioner’s expert’s theory for Claim 15 is defective because it is premised on misconstruing the term “*signal*” in the “*edit mode entry signal*” as meaning “an indication.” Houh Decl., ¶ 362. I note that the source of this definition is a 2024 online definition of “signal” from Merriam-Webster’s:

The screenshot shows the Merriam-Webster online definition for the word "signal". At the top, the word "signal" is displayed in a large, bold, black serif font, followed by "1 of 3" and the word "noun" in a smaller, blue sans-serif font. Below this, the word "sig-nal" is shown with a phonetic guide and a speaker icon. A link for "Synonyms of signal" is provided. The definition is presented in a numbered list format. Item 1 is ": SIGN, INDICATION". Item 2 is "a : an act, event, or watchword that has been agreed on as the occasion of concerted action" and "b : something that incites to action". Item 3 is ": something (such as a sound, gesture, or object) that conveys notice or warning". Item 4 is "a : an object used to transmit or convey information beyond the range of human voice" and "b : the sound or image conveyed in telegraphy, telephony, radio, radar, or television". The final item, "c : a detectable physical quantity or impulse (such as a voltage, current, or magnetic field strength) by which messages or information can be transmitted", is highlighted with a yellow background and enclosed in a blue rectangular border.

signal 1 of 3 noun

sig-nal 'sig-nəl

[Synonyms of signal](#)

- 1 : SIGN, INDICATION
- 2 a : an act, event, or watchword that has been agreed on as the occasion of concerted action
b : something that incites to action
- 3 : something (such as a sound, gesture, or object) that conveys notice or warning
- 4 a : an object used to transmit or convey information beyond the range of human voice
b : the sound or image conveyed in telegraphy, telephony, radio, radar, or television
c : a detectable physical quantity or impulse (such as a voltage, current, or magnetic field strength) by which messages or information can be transmitted

Ex. 1012 (annotated in blue, highlighted).

385. A POSITA would not understand that a “*edit mode entry signal*” of the ’286 Patent transmitted by the client “*over the network*” is “an indication.” I note that the Petitioner and its expert disregarded the definition that is pertinent to the ’286 Patent, which is “a detectable physical quantity or impulse ... by which messages or information can be transmitted.” Ex. 1012 (definition 4.c). I do not believe a special construction is needed for “*signal*,” but it most certainly does not mean “an indication.” This is another reason why the Petitioner’s and its expert’s obviousness theory for Claim 15 is defective.

386. The Petitioner’s expert references a statement in the ’286 Patent that “the client may enter the cue point mode in any useful way” including clicking on an image, selecting a GUI control, or choosing a spot on video progress bar. See Houh Decl., ¶ 363. However, Claim 15 does not claim entering the *cue point edit mode* in any possible way. Claim 15 recites entering the cue point edit mode in response to receiving from the client, “*over the network*” an “*edit mode entry signal*” that triggers the “*video editing environment*” to switch to the “*cue point editing mode*.”

387. Claim 15 is supported by the following passage from the ’286 Patent, which the Petitioner and its expert fail to acknowledge:

In some embodiments, the method further provides a cue point editing mode configured to perform at least one of creating, modifying, deleting, or interpreting at least a portion of the set of cue point information. In certain of these embodiments, the method further **receives a pause signal over the network** representing that playback of the client video data has been paused, and transmits [into] the cue point editing mode **after receiving the pause signal**. In certain others of these embodiments, the method further receives a pause signal over the network representing that playback of the client video data has been paused, receives a time position representing the time position within the client video data at which play back was paused, and creates a cue point location associated at least in part with the client video data and the time position.

'286 Patent, 3:48-61.

388. Therefore, consistent with Claim 15, the specification discloses transitioning into the *cue point editing mode* in response to the *VEE* receiving from the client a “*edit mode entry signal*” transmitted “*over the network.*” There is no such disclosure or suggestion in Vasilevsky, as explained above.

389. Thus, the combination of Vasilevsky and Logan does not meet the additional limitation of Claim 15.

G. Vasilevsky and Logan Fail to Meet Claim 17 for *Generating, Automatically, By Using the Computer System ... A Portion of the Set of Cue Point Information*

390. The Petitioner asserts that Claim 17 is obvious based on Vasilevsky and Logan. Pet. (IPR2024-01101), 86-87. Claim 17 is patentable over the references for at least the same reasons set forth for independent Claim 13.

H. Vasilevsky and Logan Fail to Meet Claim 18 for *Receiving, Over the Network, Authentication Information Related to Access Authority of the Remote Client Terminal*

391. In Claim 18, the remote client terminal that transmits the cue point editing command to cause modifying a portion of the cue point information at the server side according to Limitations 13.4-13.5 has “*authentication information relating to access authority*” that is received over the network at the server-side.

392. For limitation 13.4 (“*cue point editing command*”), the Petition asserts:

To edit a bookmark, client 124 enters an edit mode by selecting a bookmark, which displays the interface of Figure 6 showing a text box “includ[ing] such information as the bookmark establisher’s name, a title of the bookmark, and a brief description.” EX1008, 22:8-15. Client 124 can “modify all of the information,” **resulting in a command to edit the bookmark being sent from the client 124 and received using the centralized media server.**”

Pet., 83 (emph. added).

393. For Claim 18, the Petitioner refers to claim 5 of Vasilevsky providing that “said media server is adapted to enable a particular program bookmark to be edited only by a user purporting to have established the particular bookmark.” Pet., 88. The Petitioner’s expert cites Vasilevsky at 20:1-10 and 21:9-20 for the same

proposition. Houe Decl., ¶ 367. But the mere recitation of functionality that the bookmark can only be edited by the creator of the bookmark does not teach or suggest that the server-side of the Vasilevsky-Logan combination receives “*over the network*” “*authentication information relating to access authority*” for the client editing the bookmark. As discussed in Section VIII.G for Gupta, such authentication information could easily be received by the server-side of Vasilevsky-Logan by direct input and other manners other than over the network. Accordingly, Vasilevsky-Logan does not teach or suggest Claim 18. See Pet., 88; Houh Decl., ¶¶ 365-368.

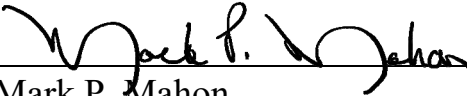
I. Claims 14-15 and 17-20 Are Patentable Over Vasilevsky and Logan

394. Claims 14-15 and 17-20 are patentable over Vasilevsky and Logan for at least the same reasons set forth for Claim 13.

XIV. CONCLUSION

395. I hereby declare that all statements made herein of my own knowledge are true and all statements made herein on information and belief were and are believed by me to be true, and that all statements herein were and are made with the knowledge that willful 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 and that any such willful false statements may jeopardize the validity of the application or any patents issued thereon.

Respectfully submitted,


Mark P. Mahon

Dated: February 12, 2025