

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AYLO FREESITES LTD,
Petitioner

v.

WELLCOMEMAT, LLC,
Patent Owner

IPR2024-00710
U.S. Patent No. 8,307,286

REPLY DECLARATION OF HENRY HOUH, PH.D.

TABLE OF CONTENTS

I.	Introduction.....	1
II.	The '286 Patent's Video Editing Environment	4
A.	A POSITA Would Have Understood the '286 Patent to Describe a Server-side “video editing environment” (VEE) that Receives Cue Point Information From a Client Via a Web Browser Form.....	4
B.	The '286 Patent Uses Conventional Web Server Technology Including CGI and PHP.....	25
III.	Ground 1 (Gupta).....	28
A.	Claim 13	28
1.	Gupta Discloses a Server-Side VEE That Receives a Set of Video Information	28
2.	Gupta Discloses Server-side Software For Editing Cue Points That are Created/Modified at the Server using Cue Point Information and Parameters Received From a Client	30
3.	Gupta Discloses the Same Server-Side Methodology for “Creating”/“Modifying” a Cue Point as Described in the '286 Patent	36
B.	Claim 15	40
C.	Claim 18	41
IV.	Ground 2 (Fegan).....	42
A.	Claim 13	42
1.	Fegan Discloses Server-Side Segmenting/Annotation Functionality	42
2.	Fegan's Segmenting/Annotation Tools are Distinct from the Media Player and are Server-side Functionality.....	49
3.	Fegan's Media Player Operates in the Same Manner as the '286 Patent's Sole Media Player Example	55
4.	Fegan Discloses the Same Server-Side Methodology for “Creating”/“Modifying” a Cue Point as Described in the '286 Patent	58

5.	Fegan Discloses Limitations [13.4]-[13.5]	66
B.	Claim 15	70
C.	Claim 18	73
V.	Ground 5 (Vasilevsky in view of Logan)	74
A.	Claim 13	74
1.	A POSITA Would Have Been Motivated to Modify Vasilevsky’s HAN to be a Public Network Such as Logan’s Internet	74
2.	Vasilevsky’s Server Components Satisfy the Claimed VEE	81
B.	Claim 15	83
C.	Claim 18	83
VI.	Conclusion	84

I. Introduction

1. My name is Henry Houh, and I have been retained by counsel for Aylo Freesites, Ltd (“Petitioner”) to provide my opinions regarding the patentability of claims 13-15 and 17-20 of U.S. Patent No. 8,307,286 (“’286 Patent”) (EX1001).

2. In prior declarations submitted in this proceeding and in IPR2024-01101, I have set forth my opinions concerning claims 13-15 and 17-20 of the ’286 Patent. EX1002 (“Original Declaration”); IPR2024-01101, EX1002.¹

3. I have been asked to prepare this Reply Declaration to present my opinions in response to arguments made by Patent Owner WellcomeMat LLC (“WellcomeMat”) in its Patent Owner Response (“POR”) and to opinions expressed in the Declaration of Dr. Mark Mahon (EX2015) (“Mahon Declaration” or “Mahon Decl.”).

4. I am being compensated at my normal consulting rate of \$650 per hour for my time. My compensation is not contingent on the outcome of this proceeding, or of any proceedings relating to the ’286 Patent.

¹ I have been informed that IPR2024-01101 has been consolidated into this proceeding. Unless otherwise indicated, references to “Original Declaration” or “EX1002” refer to my declaration in this proceeding—IPR2024-00710.

5. In forming my opinions set forth herein, I relied upon my education and experience, which I describe in my Original Declaration, in my *curriculum vitae* (EX1003), and in my deposition (EX2035). As noted in my Original Declaration, I assumed a priority date of May 7, 2006—the earliest date of a provisional application. EX1002, Section VI.C. I reserve the right to argue for a later priority date in other proceedings.

6. I have considered the following:

- Materials listed in my Original Declaration (EX1002, ¶23), particularly the '286 patent (EX1001), U.S. Patent Application Publication No. 2005/0081159 (“Gupta”) (EX1004), U.S. Patent Application Publication No. 2005/0144305 (“Fegan”) (EX1005), WO 2003/067594 A1 (“Vasilevsky”) (EX1008), U.S. Patent Application Publication No. 2003/0093790 (“Logan”) (EX1009), and the File wrapper for the '286 Patent (EX1010);
- Board’s Institution Decision (Paper 13) (“Institution Decision”);
- Patent Owner’s Response (Paper 24) (“POR”);
- Declaration of Dr. Mark Mahon (EX2015) (“Mahon Declaration” or “Mahon Decl.”);
- Deposition Transcript of Dr. Mark Mahon, March 13, 2025 (EX1024) (“Mahon Dep.”);

- Tanenbaum, COMPUTER NETWORKS (2003) (EX2025)
 (“Tanenbaum”);
- WO 2005/122025 A2 to Krikorian (Sling Media, Inc.) (EX2032)
 (“Krikorian”);
- Excerpts from Laura Lemay, SAMS TEACH YOURSELF WEB
 PUBLISHING WITH HTML 4 IN 21 DAYS (2nd ed.) (2000) (EX1025)
 (“Lemay”);
- Excerpts from David Gourley and Brian Totty, HTTP: THE
 DEFINITIVE GUIDE (2002) (EX1026) (“Gourley”);
- Wayback Machine archive of Microsoft Windows Server 2003,
 Internet Information Services,
 <http://www.microsoft.com/WindowsServer2003/iis/default.mspx>
 (archived Oct. 27, 2004) (EX1027);
- Microsoft Learn, ISAPI Extensions,
 [https://learn.microsoft.com/en-us/previous-versions/iis/6.0-
 sdk/ms525282\(v=vs.90\)](https://learn.microsoft.com/en-us/previous-versions/iis/6.0-sdk/ms525282(v=vs.90)) (EX1028);
- Excerpts from Simon Collin, SETTING UP A WEB SERVER (1997)
 (EX1029);

- Wayback Machine archive of Wikipedia, DOCSIS, <https://en.wikipedia.org/wiki/DOCSIS> (archived Feb. 9, 2006) (EX1030);
- Xfinity Support, Upgrade internet devices impacted by expiring DOCSIS certificates, <https://www.xfinity.com/support/articles/docsis-certification> (EX1031); and
- Excerpts from Ellen Finkelstein and Gurdy Leete, MACROMEDIA FLASH 8 FOR DUMMIES (2006) (EX1032).

7. My understanding of the legal principles at issue are set forth in my Original Declaration. EX1002, ¶¶28-40.

II. The '286 Patent's Video Editing Environment

A. A POSITA Would Have Understood the '286 Patent to Describe a Server-side “video editing environment” (VEE) that Receives Cue Point Information From a Client Via a Web Browser Form

8. Dr. Mahon opines in his declaration that the “video editing environment” (“VEE”) is not met by the prior art’s disclosure of server-side software that receives cue point information submitted from a client via a web form. Mahon Decl., ¶¶99-106 (Dr. Mahon opining that the client-side UIM in Gupta “creates” and “edits” annotations and that the server side receives those created/edited annotations and stores them into memory), ¶¶186-206 (Dr. Mahon opining that the Fegan’s

client-side browser creates/edits annotations while the remote server is used to save and store the annotations).

9. As I understand Dr. Mahon's opinions in his declaration, Dr. Mahon believes that such an arrangement (where the server side receives cue point information submitted from a client via a web form) is a *client-side* VEE rather than a *server-side* VEE, as he believes a cue point is "created" or "edited" at the client side under this arrangement.

10. I disagree with Dr. Mahon. In my opinion, Dr. Mahon's position is unsustainable at least because the '286 Patent describes an example of a VEE receiving cue point information via a client web browser form, and specifically refers to such instance of a VEE as *server-side*.

11. To start, the '286 Patent repeatedly describes the use of a client web browser, including such use in the particular context of presenting the VEE user interface to a client user.

FIG. 5 provides an exemplary flow diagram summarizing methods relating to client terminal interactions with video editor environments in various embodiments of the invention. The method 500 may begin when a client enters a VEE 510. **The client may enter the VEE 510 by** executing an application invoking various software and/or hardware commands, **accessing a webpage**, or in any other useful way.

'286 Patent, 24:56-62 (emphasis added).

The method 300 may then generate a VEE graphical user interface (GUI) 316.

'286 Patent, 13:48-49 (emphasis added).

It will be appreciated that the GUI may be generated 316 in any useful way for providing a useful interface to the VEE. For example, **the GUI may be generated 316 as a webpage on a client's web browser, as a frame within a webpage**, as a dedicated client-level application on the client's terminal (with the VEE itself running either at the client- or server-level), or as a background script to be executed by another application (e.g. as Visual Basic commands for an interpreter which may use those commands to generate the GUI).

'286 Patent, 13:59-67 (emphasis added).

For example, **the VEE may load as** a script in the background of a **webpage**, as a server-level application, as a client-level application, or with various components of the application loading in different ways.

'286 Patent, 13:3-6 (emphasis added).

In certain embodiments, the GUI elements will be embedded within the context of another environment (e.g. **webpage**, application, etc.).

'286 Patent, 14:8-13 (emphasis added).

26. The system of claim 21, wherein the application server is configured to run the video editing module over the public network, so as to provide the edit interactivity according to data received over the public network via a standard Internet browser running on the remote client terminal.

'286 Patent, claim 26.

12. Dr. Mahon also confirmed during his deposition that the '286 Patent encompasses web browser-based access for the VEE.

Q So I'd like to turn back to Figure 1A of the '286 patent, Exhibit 1001.

A I'm going to the figure. Let me just zoom out of it.

Okay. I'm there.

Q All right. And prior to the break, do you recall that we were discussing the VEE 116?

A Yes.

Mahon Dep., 78:17-2.

Q Could the VEE be implemented as a website or web page?

A You could deliver the capabilities for the environment via access through web access if you wanted to. But the actual capabilities would have to be -- have underlying software application to deliver that functionality.

Q And what underlying software application would be involved, in that case, of web access?

A It depends on how the designer wants to implement it. But, for example, if you're talking about a particular use case, they could use a -- any one of a number of streaming protocols to deliver streaming video or video to the user. So one of those protocols could be used after the user initially accesses, in your -- your hypothetical, via the -- the -- an HTTP request or a web page request.

Q So just so I understand you correctly. The VEE could be implemented as -- as web access based on a software application?

A Well, this is a client server architecture, which definitely means you're accessing it via the Internet, which colloquially means via the web, to access the capabilities. So, by definition, it's delivered as a web capability.

Q So in that instance, the client accesses the VEE via a web page?

A No. I didn't say that. I said the capability is delivered via the web. You could use a browser, or you can have a particular deployed application that reaches out to the VEE and delivers the content to that application on the client. There is various ways to do it.

Q If I understand you correctly, that one -- one way is to use a browser to access the VEE?

A What I said was, you can use a browser to access the client server architecture. That is, in this case, the server capabilities. And then the underlying capabilities of that server, whatever

service they're providing you, the user would utilize those services to request access to the VEE.

Mahon Dep., 79:14-81:15.

Q So that's a “yes,” that a web server responding to HTTP request is encompassed by the '286 patent?

A The '286 is disclosing a client server architecture, and in the client server architecture, certainly, you could request access to the server's capabilities and which services they're providing via initial HTTP request or connection. But after that, obviously, you can implement it in various ways following.

Mahon Dep., 87:12-22 (objection omitted).

13. I also understand that WellcomeMat's Patent Owner Response acknowledges that claim 13 of the '286 Patent encompasses a VEE being accessed by a client web browser.

The claim encompasses the use of a browser to access a server-side VEE that performs editing of cue points per limitation 13.1. For example, the '286 Patent discloses that the server-side VEE cue point editing can be controlled by a client device using a browser. See '286 Patent, 13:58-67.

POR at 38 (footnote 1).

14. Additionally, the '286 Patent repeatedly describes the use of web browser forms for client users to input cue point information. The '286 Patent

describes that the webpage displayed in the client's web browser includes a web form for creating and editing cue point information. A user inputs cue point information into the form at the client side, and that information is sent to the server side for storage in cue point server 114. The '286 Patent also describes that cue point server 114 may be implemented using a database, which a POSITA would have understood to involve database entries that are added/modified.

When the method 370 determines that full interactivity will be provided 376-2, the method 370 may generate a **cue point information form 382**. Generating the cue point information form 382 may involve any process by which the method 370 may **solicit** the desired or required **cue point data from a client**. It will be appreciated that in different cases, different types of cue point information may be desired or required **as part of** the **cue point creation or modification**.

In some embodiments, the method 370 may open a new window, frame, or portion of the GUI which may provide various fields and/or controls relating to different types of cue point information. The client would then be able to **input cue point information into the form** by using some or all of these fields and/or controls. Further, the fields and/or controls may be tailored or may have restrictions based on the associated type of cue point information. In one example, a **client** may **enter the cue point name in a text string field** which allows a maximum length of thirty characters. In another example, a client may

select a cue point name from a list of common or acceptable cue point names provided in a drop-down list. In yet another example, the client may provide the cue point name by saying the name into a microphone (after which point the method 370 may use various voice-recognition algorithms to convert the audio into a textual representation of the cue point name).

'286 Patent, 21:62-22:19 (emphasis added).

In some cases, the method 370 may determine that partial interactivity will be provided 376-3. In these cases, the method 370 may detect and/or load some of the desired or required cue point properties 380-2 in any effective way from any useful source. After detecting or loading some cue point properties 380-2, the method 370 may **generate a cue point information form** 384 to **solicit** the remaining **desired or required cue point information from a client**.

Whether the method 370 generates a **full cue point information form 382** (the method 370 determines that full interactivity will be provided 376-2) or a **partial cue point information form 384** (the method 370 determines that partial interactivity will be provided 376-2), the method may then **receive cue point information from a client**. The client may be any effective provider of the desired or required cue point information, including a human client of a client terminal, a computer (e.g. automated or artificial intelligence) client of a client terminal, an application running compatible and interfacing processes or functions, or any other information provider.

It will be appreciated that there are many ways to solicit **input** from the **client** for this purpose. For example, there are many types **of GUI fields and controls (e.g. text fields, radio buttons, sliders, check boxes, drop-down menus, etc.)**, many types of input devices (e.g. keyboard, mouse, microphone, scanner, optical character recognition system, writing tablet, computer script, etc.), and many types of potentially editable cue point information. Some or all of these options may be provided or limited, possibly based on various factors, including a component's technical capabilities, a client's permissions, or a cue point's type.

'286 Patent, 22:20-50 (emphasis added).

In some embodiments, the method 370 may open a new window, frame, or portion of the GUI which may provide various fields and/or controls relating to different types of cue point information. The client would then be able to input cue point information into the form by using some or all of these fields and/or controls.

'286 Patent, 22:3-8.

Whether the method 370 generates a full cue point information form 382 (the method 370 determines that full interactivity will be provided 376-2) or a partial cue point information form 384 (the method 370 determines that partial interactivity will be provided 376-2), the method may then receive cue point information from a client. The client may be any effective

provider of the desired or required cue point information, including a human client of a client terminal, a computer (e.g. automated or artificial intelligence) client of a client terminal, an application running compatible and interfacing processes or functions, or any other information provider.

'286 Patent, 22:28-39.

15. The '286 Patent provides a specific example (passage reproduced below) of a VEE that receives cue point information provided by a client via web form input, and explicitly refers to such a VEE as being *server*-side. In this passage, the '286 Patent discloses that cue point information is transmitted via the client-side web form to the VEE, and is received by the VEE which saves the received cue point information into server-side databases. In particular, in this passage, the '286 Patent describes that the cue point information form is “displayed on a client terminal” while the cue point editor (*i.e.*, the VEE) “resides on a *server* terminal.”

After the method 370 receives all the desired or required cue point properties, the method 370 may generate and save the cue point 392. In some embodiments, the method may ask for verification from a client before generating and/or saving the cue point 392. In other embodiments, the method 370 may only **generate and/or save the cue point after the client submits the information.** For example, **the cue point information form** may be displayed on a **client terminal** while the **cue point editor** resides on a **server terminal**. In this example, the client may be

required to submit the form over a network to the server computer before the cue point editor has the information it needs to generate the cue point.

Further, the method 370 may save the cue point in any effective way in any location. For example, the method 370 may perform a database insert function to insert the new or modified cue point information into an existing cue point database, which may be stored on a server.

'286 Patent, 23:12-28 (emphasis added).

16. The above passage also refers the client “submit[ing] the form over a network to the server computer before the cue point editor has the information it needs to generate the cue point.” In other words, the cue point is “generated” after the server side receives the form information inputted at the client.

17. Furthermore, the passage above refers to “generat[ing]” and “sav[ing]” the cue point “after the client submits the information.” Based on these descriptions in this passage, a POSITA would have understood this passage as describing (i) cue point information transmitted via the client-side web form to the VEE and received by the VEE, with the VEE generating/saving a cue point based on the form data, and (ii) the VEE being a server-side VEE.

18. As I mention above, the '286 Patent further describes the storage of the cue points at the server side being a database in the cue point server 114.

During or after cue point information is received 324, the method 300 may **update cue point associations** 326. Updating cue point associations 326 may comprise **storing, modifying, removing, or otherwise processing information stored in relation to cue points**, video files, or other data. In one example, a client may choose to save new cue point information, at which point the method 300 may **save the information to a relational database stored on a cue point server** with an attribute pointing to the loaded video file. In another example, the name of the cue point may be saved to a database which records cue point names and the number of times those (or similar) names occur within a set of video files.

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. The final updated information may then be saved to databases or other, more

permanent, storage locations when the client uploads the information or exits the VEE.

'286 Patent, 16:12-41 (emphasis added).

The application server 110 may also be in communication with one or more cue point servers 114, which may **store and/or serve cue point information 115**, or pointers to cue point information 115. In one example, the cue point server 112 may be a dedicated computer or network of computers with the capability to store and index sets of cue point information 115.

'286 Patent, 7:42-48 (emphasis added).

19. In my opinion, Figure 6 of the '286 Patent illustrates an example of the user interface that may be used in the example described in the passage reproduced above (of the '286 Patent, 23:12-28). The '286 Patent describes Figure 6 as illustrating “an exemplary screen shot of an embodiment of a video editing environment **as viewed by a client terminal**.” '286 Patent, 5:53-55, 25:64-66 (emphasis added). As seen in this figure, the user interface includes form fields for user input of cue point information. I reproduce Figure 6 below, and add color annotations as I discuss further below.

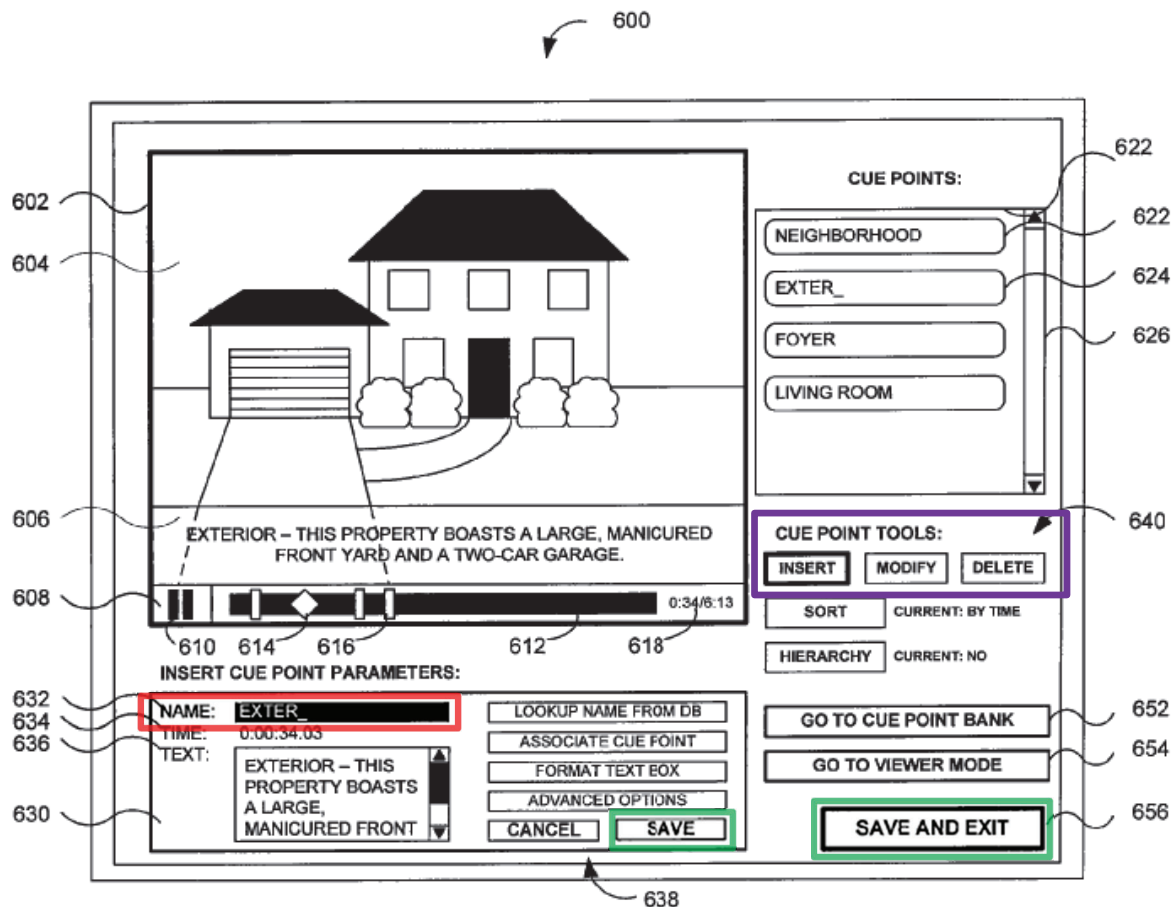


Fig. 6

'286 Patent, Fig. 6 (annotated).

20. As seen above and based on the quoted portions below, Figure 6 shows a client user interface that contains cue point tools 640 (annotated above in purple). A user can create/modify a name to be associated with a cue point by editing the “Name” field 632 (annotated above in red). The '286 Patent even describes that in this figure, a user is “in the middle of entering the cue point name ‘EXTERIOR’” in the Name field 632.

The cue point editor screen 600 may also comprise other areas, like a **cue point tools area 640**. The cue point tools area 640 may comprise various fields and controls for editing cue points. For example, as illustrated, the cue point tools area 640 comprises virtual **buttons for inserting, modifying, and deleting cue points**.

'286 Patent, 27:60-65 (emphasis added).

The client then began entering information into the cue point editor. As shown, the client has already entered the comment text associated with the cue point (see the "Text" field 636 and the comment area 606). **The client is currently in the middle of entering the cue point name "EXTERIOR," such that it currently reads "EXTER_" (the "_" representing a cursor location for the next character to be entered) (see the "Name" field 632)**. Also, as shown, the name of the cue point in the cue point listing area 620 is being automatically updated substantially in real time to match the name being entered in the "Name" field 632.

'286 Patent, 28:50-60 (emphasis added).

21. As seen above and based on the quoted portions below, Figure 6 shows "Save" and "Save and Exit" buttons (annotated above in **green**), which cause the interface to transmit to the server all cue point information entered by the client. The '286 Patent describes that user selection of the "Save" button causes saving and

submission of the inputted cue point information, and also describes that a “save” refers to storing the information on the cue point server’s database.

Additionally, as illustrated, various virtual buttons are provided, including a "Lookup Name from DB" button for allowing a client to look up an appropriate cue point name from a list of cue point names stored in a database; an "Associate Cue Point" button for allowing a client to associate the cue point being edited with another video file or another cue point (e.g. in a hierarchy); a "Format Text Box" button for allowing a client to view more options for formatting the text and other attributes of the comment area 606; an "Advanced Options" button for allowing a client to view advanced options for cue point editing, GUI display, etc. (e.g. for changing the icon associated with a viewpoint, changing a thumbnail associated with a cue point or video file, changing the size and shape of cue point editing screen 600 elements, etc.); a "Cancel" button for allowing a client to cancel the current cue point editing; and a **"Save" button for allowing a client to save the current cue point editing and possibly submit the cue point information.**

’286 Patent, 27:40-59 (emphasis added).

In one example, a **client** may choose to **save new cue point information**, at which point the method 300 may **save** the information **to a relational database stored on a cue point server** with an attribute pointing to the loaded video file. In another example, the name of the cue point may be saved to a

database which records cue point names and the number of times those (or similar) names occur within a set of video files.

'286 Patent, 16:15-23 (emphasis added).

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. The **final updated information may then be saved to databases** or other, more permanent, storage locations **when the client uploads the information** or exits the VEE.

'286 Patent, 16:24-41 (emphasis added).

In other embodiments, the method 370 may only **generate** and/or **save** the cue point **after the client submits the information**. For example, the cue point information **form** may be displayed on a client terminal while the **cue point editor resides on a server terminal**. In this example, the client may be required to **submit**

the form over a network to the server computer before the cue point editor has the information it needs to generate the cue point.

'286 Patent, 23:16-23 (emphasis added).

22. The type of web forms disclosed in the '286 Patent and shown in Figure 6 were notoriously well-known to a POSITA long-before the earliest priority date of the '286 Patent. For example, "Sams Teach Yourself Web Publishing with HTML 4 in 21 Days, Second Edition" by Lemay (EX1025) ("Lemay"), published in 2000, describes the same form-based web framework disclosed in the '286 Patent including Figure 6. Lemay explains that "HTML forms have been around virtually since the beginning of the HTML language, and are widely supported by all leading web browsers." Lemay at 429. Lemay explains that such web forms "enable you to gather just about any kind of information for immediate processing by a server-side script or for later analysis using other applications." *Id.*

23. Lemay further explains the known aspects of forms as described in the '286 Patent, including that "a form is part of a Web page that you create using HTML elements" with "special controls, such as buttons, text fields, check boxes, submit buttons, and menus" which "make up the user interface for the form." Lemay at 430 (emphasis added). Lemay explains that after a user enters information into the form and "clicks the Submit button," server-side scripts such as Common Gateway

Interface (“CGI”) that are “resident on the Web server” are executed to “take the data and store it.”

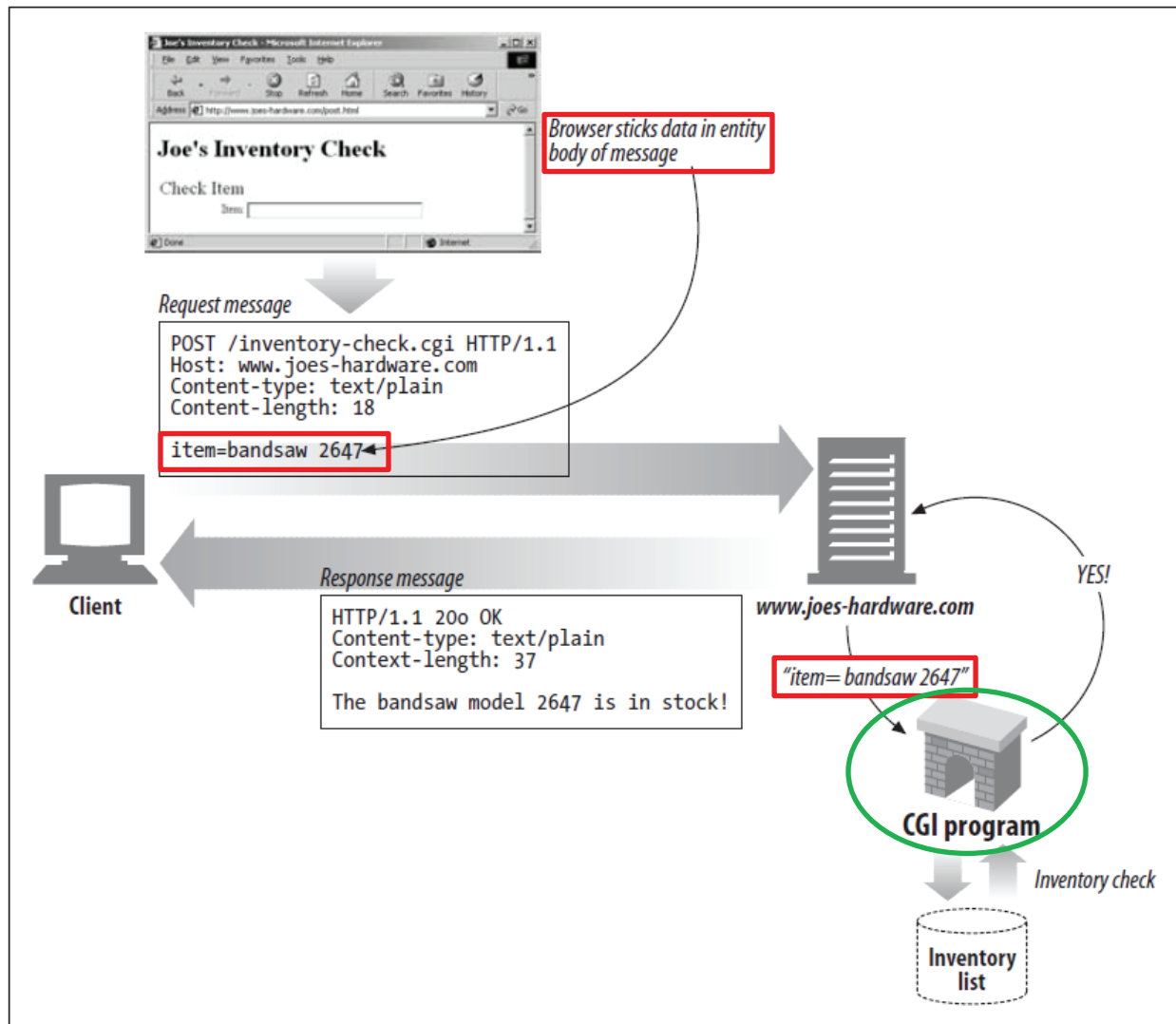
An HTML form allows you to gather information from a person visiting your Web page. You can take the information you collect and send it to a **script residing on your Web server** to manipulate or **store it**.

When a person fills out an HTML form, he or she enters information or makes choices using the form controls. The final step is submitting it, at which time several things happen. First, the form identifies the controls within the form that contain data and builds a form data set to contain it. Next, the data set is encoded and sent to the Web server to be processed.

It is very important that you understand the implications of this final step. The data, after all, is what you are after! This is the reason you've chosen to create a form in the first place. After a user clicks the Submit button, the process ceases to be one of pure HTML and becomes reliant upon scripts (called **Common Gateway Interface, or CGI scripts**, most often written in a scripting language called Perl) that are **resident on the Web server**. In other words, for your form to be successful, you **must already have a script on the server** that will take the data and **store** it or manipulate it in some manner.

Lemay at 430 (emphasis added).

24. Additionally, “HTTP: The Definitive Guide” by Gourley (“Gourley”) (EX1026), published in 2002, further explains how a POSITA would have understood the above form example from the ’286 Patent as executing server-side CGI or similar software to receive and manipulate and/or store information sent from the client web browser over the Internet using HTTP. Gourley explains that “CGI is an early, simple, and popular interface for executing server-side applications.” Gourley at 123. Gourley explains that often the HTTP POST method is used “to support HTML forms” such that “data from a filled-in form typically is sent to the server, which then marshals it off to where it needs to go (e.g., to a server gateway program, which then processes it).” Gourley at 55. Gourley presents an exemplary Figure 3-10 (annotated below) that shows “a client making an HTTP request—sending form data to a server—with the POST method” where it is received by the server-side CGI program to process the information. Gourley at 55-56.



Gourley at 56 (Figure 3-10) (annotated).

25. Thus, a POSITA would have possessed such knowledge of web forms and would have understood '286 Patent Figure 6's "Save" and "Save-and-Exit" buttons to provide such functionality and operate in this manner. The '286 Patent plainly discloses using forms provided on the client user interface, with a "Save" or "Submit"-type button which triggers transmission of the entered information to the

server side which is handled by the server-side software for creating/modifying cue point information at the server-side.

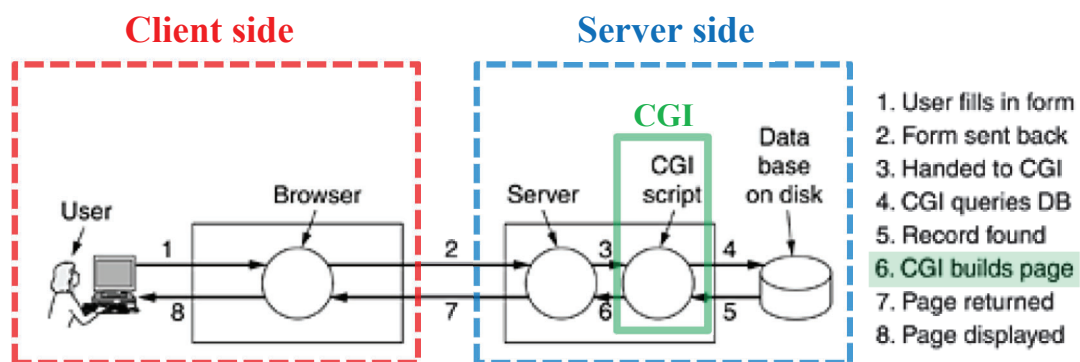
B. The '286 Patent Uses Conventional Web Server Technology Including CGI and PHP

26. The '286 Patent is very broad and open-ended in its disclosures, does not purport to limit itself to any particular implementation or client-server configuration. In fact, the '286 Patent discloses that “many types [of] video editing environments are possible” and the “descriptions of embodiments relating to video editing environments in FIGS. 3-7 should not be construed as limiting the scope of the invention.” '286 Patent, 29:29-35. Similarly, Dr. Mahon testified when deposed that the '286 VEE could be implemented in a variety of ways, and ultimately it's the designer's choice to choose a particular implementation. Mahon Dep., 79:6-86:1.

27. According to the '286 Patent, the VEE may employ known, conventional server-side technologies such as “CGI [Common Gateway Interface] scripts.” '286 Patent, 18:15-17. As I explained above in Section II.A, conventional server-side CGI, or some similar server-side script, must be present on the server in order to receive, process, and store information submitted by a client using a form on a webpage. This would have been very well-known and understood by any POSITA before the earliest priority date of the '286 Patent.

28. Additionally, Tanenbaum's COMPUTER NETWORKS (EX2025), which WellcomeMat actually submitted in this proceeding, likewise describes the

operation of CGI and similar server-side scripts. Tanenbaum explains that CGI is “[t]he traditional way to handle [web] forms and other interactive Web pages,” such as those disclosed in ’86 for modifying and creating cue point information. Tanenbaum at 496. Tanenbaum presents “an example of how CGI often works,” where a user submits form data via a web browser to create a new entry in the server-side database. Tanenbaum at 496. Tanenbaum explains that in this example, a “user fills in the form and clicks on *submit*”, causing the information to be sent to the server side. *Id.* The server-side CGI script “pares the [form] parameters, **makes an entry in the database** for the new customer, and sends back an HTML page” to the client web browser as the next user interface screen. *Id.*



Tanenbaum at 496 (Fig. 7-33 (annotated in color))

Server-Side Dynamic Web Page Generation

To see why server-side content generation is needed, consider the use of forms, as described earlier. When a user fills in a form and clicks on the submit button, a message is sent to the server

indicating that it contains the contents of a form, along with the fields the user filled in. This message is not the name of a file to return. What is needed is that the message is given to a program or **script to process**. Usually, the processing involves using the user-supplied information to **look up a record in a database on the server's disk** and **generate a custom HTML page to send back to the client**.

Tanenbaum at 495-96 (emphasis added).

29. This client-server action described in Tanenbaum mirrors the disclosures of the '286 Patent, which describes a client transmitting cue point information using a web form to the application server, which saves such information in the cue point server database. *See* Section II.A above; *see also* Mahon Decl., ¶134 (where Dr. Mahon acknowledges that HTTP requests are “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**)” (emphasis added)).

30. Tanenbaum also explains that there exist other conventional server-side dynamic web page generation techniques comparable to CGI, including PHP (*i.e.*, “PHP: Hypertext Preprocessor”). Tanenbaum explains that PHP scripts are executed by the server to generate dynamic HTML pages to serve to the client.

CGI scripts are not the only way to generate **dynamic content** on the server side. Another common way is to embed little scripts

inside HTML pages and have them be **executed by the server itself** to **generate the page**. A popular language for writing these scripts is **PHP (PHP: Hypertext Preprocessor)**. To use it, the server has to understand PHP (just as a browser has to understand XML to interpret Web pages written in XML). Usually, servers expect Web pages containing PHP to have file extension php rather than html or htm.

A tiny PHP script is illustrated in Fig. 7-34; it should work with **any server that has PHP installed**.

Tanenbaum at 496 (emphasis added); *see also id.* at 496-98.

31. A POSITA would have understood the '286 Patent to draw from these conventional web technologies including CGI and PHP, given their prevalent use for serving web pages. Moreover, the '286 Patent does not offer any other specific implementational details for web technology (let alone for the form functionality described in the '286 Patent), and therefore such absence would have led a POSITA to understand that conventional web technologies are being utilized including for the forms.

III. Ground 1 (Gupta)

A. Claim 13

1. Gupta Discloses a Server-Side VEE That Receives a Set of Video Information

32. Dr. Mahon opines that Gupta does not disclose receiving a set of video information at the VEE as claimed. Mahon Decl., ¶¶154-58. I disagree.

33. As I explained in my Original Declaration (EX1002, Sections X.B.1-3), Gupta explicitly describes that streaming media server 11 accesses streaming media content that may be stored in a separate file storage system. Gupta, ¶[0030] (“stored ... in a database or other file storage system”). In this scenario, streaming media server 11 receives the videos from the database/file storage system when they were originally stored in the first place or during access.

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. Some media streams might be stored as files in a database or other file storage system, while other media streams might be supplied to the server on a "live" basis from other data source components through dedicated communications channels or through the Internet itself.

Gupta, ¶[0030] (emphasis added).

34. Dr. Mahon’s declaration appears to acknowledge Gupta’s reference to storage of media streams “in a database or other file storage system” (Mahon Decl., ¶155) but does not address why such teaching of the media server 11 receiving video information from such other databases or file storage systems does not satisfy the claimed recitation of receiving video information.

35. Additionally, Gupta discloses media server 11 receiving “live” media streams “supplied” to it “from other data source components.” Gupta, ¶[0030]. Dr. Mahon opines that even in this scenario of live streams, “there is no disclosure that the media stream is being received so that its cue point information (e.g., annotations) can be edited.” Mahon Decl., ¶156. I disagree.

36. Gupta discloses that its invention “relates to creating, viewing, and temporally positioning annotations for media content” (Gupta, ¶[0003]), including for “streaming” media. Gupta, ¶[0033] (“annotations managed by annotation server 10 correspond to the streaming media”). Dr. Mahon’s suggestion that no annotation takes place in the case of live streams, simply ignores the entire objective and purpose of Gupta.

2. Gupta Discloses Server-side Software For Editing Cue Points That are Created/Modified at the Server using Cue Point Information and Parameters Received From a Client

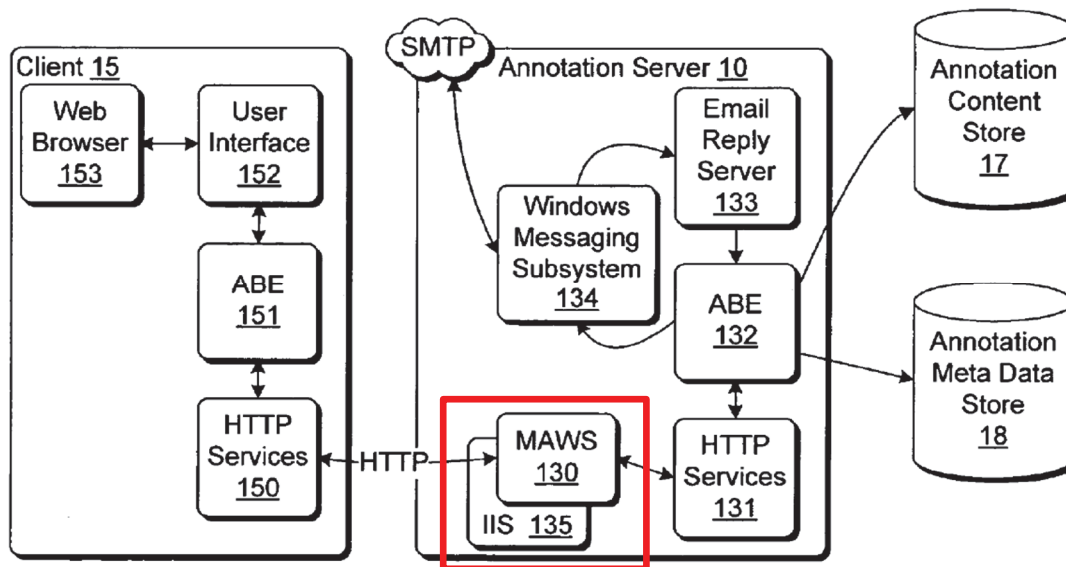
37. Dr. Mahon opines that Gupta’s system discloses client-side, rather than server-side, cue point creation and editing. *E.g.*, Mahon Decl., ¶¶95-119. I disagree.

38. As I explained in my Original Declaration, Gupta discloses software at the server-side computer system that receives video information and a set of cue point information and includes a web server that serves the web-based user interface screens to the client for display on the client web browser. EX1002, Section X.B. Gupta specifically identifies particular server-side software that produces the user

interface for editing annotations. Gupta's server-side software is running within at least media streaming server 11 and annotation server 10 that perform the functions of the claimed VEE as (i) they are part of a server-side computer system as shown in Fig. 1; and (ii) the media streaming server receives video information (*e.g.*, Gupta, ¶[0030] ("media streams might be stored as files" or "supplied to the server on a 'live' basis")) and the annotation server receives cue point information (*e.g.*, Gupta, ¶[0032] ("annotation server 10 can receive and provide annotations via direct contact with a client computer 15")).

39. Dr. Mahon opines that "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)." Mahon Decl., ¶119. I disagree.

40. Gupta actually provides details on examples of specific software forming the components of the annotation server 10, and in particular Gupta's web server functionality in providing the user interface to clients. Gupta discloses that annotation server 10 includes an "Internet Information Server (IIS) module" 135 and a "Multimedia Annotation Web Server (MAWS) module" 130, which is an "Internet Services Application Programming Interface (ISAPI) plug-in for Internet Information Server (IIS) module 135."



Gupta, Fig. 3 (annotated).

Annotation server 10 includes the Multimedia Annotation Web Server (MAWS) module 130, which is an Internet Services Application Programming Interface (ISAPI) plug-in for Internet Information Server (IIS) module 135. Together, these two modules provide the web server functionality of annotation server 10. Annotation server 10 also includes an HTTP Services module 131 which manages communication with client 15. In addition, annotation server 10 utilizes The Windows Messaging Subsystem 134 to facilitate communication with email server 13 of FIG. 1, and an email reply server 133 for processing incoming email received from email server 13.

Gupta, ¶[0047] (emphasis added).

41. I also note that Gupta, on its face, is assigned to Microsoft. A POSITA would have recognized that Gupta’s reference to “Internet Information Server (IIS)”

is referring to Microsoft's Internet Information Server (IIS) software component within the Windows Server operating system, such IIS component providing web server functionality. EX1027 (Wayback Machine archive of Microsoft Windows Server 2003, Internet Information Services). Microsoft's IIS software component was widely used at the time of Gupta for web hosting functionality.

42. A POSITA would also have recognized that Gupta's reference to the "Internet Services Application Programming Interface (ISAPI) plug-in" is to a particular software plug-in for Microsoft's IIS software component, for "developing dynamic Web content." EX1028; EX1029 at 179-80 (ISAPI is "similar to CGI;" "There is an alternative to CGI called ISAPI [Internet Server Applications Programming Interface) that works using dynamic link libraries (DLLs), so there is no need for an executable" to be executed by the Windows-based web server). A POSITA would have understood that ISAPI "provides the same functionality as CGI, but can be executed faster than CGI." EX1028.

43. These understandings of a POSITA are further reinforced by Gupta's explicit disclosure that its MAWS module 130 and IIS module 135 collectively "provide the web server functionality of annotation server 10" and therefore are responsible for generating the dynamic web pages provided to the client for display on web browser 135, including Gupta's screenshots, *e.g.*, Figures 7, 8, 21. Gupta, ¶[0047].

44. Dr. Mahon opines that Gupta’s client-side interface module 152, not Gupta’s server-side components including annotation sever 10, is the component in Gupta’s system that performs the “creating” and “editing” of annotations. Mahon Decl., ¶¶93-119. I disagree.

45. In my opinion, Gupta’s interface 152 module is not used to create or modify cue points, but is simply functionality at the client to facilitate communication between the client and the annotation server 10.

46. In fact, Gupta’s approach to providing a user interface to a client for modifying cue point information generally mirrors the server-side VEE example disclosed in the ’286 Patent, which I discussed above in Section II.A. Namely, Gupta’s web server 12 provides a user interface module 152 to a client device for communicating with annotation server 10. Gupta, ¶¶[0082] (“interface module 152 may be incorporated into the web browser ... received from ... web server 12 ... for execution at client 15”); [0081] (“interface module 152 [] presents an interface to a user (e.g., a graphical user interface (GUI)), allowing a user to make requests of annotation server 10”).

47. Gupta’s Interface module 152 facilitates the user entry of modified cue point information into a web form (e.g. Fig. 21) and sending that information to annotation server 10 for storage as a cue point in annotation content store 17. Gupta, ¶¶[0144], [0102], [0100] (“Selection of OK button 320 causes 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”), [0048] (“Annotation content authored at client 15, using user interface 152, is received by ABE 132 and maintained in annotation content store 17”).

48. I understand that Dr. Mahon does not dispute the operation that takes place in Gupta, as he confirms that ABE module 132 within annotation server 10 “receives [annotation information] from the client and stores them in the database 17” at the server side. Mahon Decl., ¶127.

49. Dr. Mahon’s contention that Gupta’s browser form-based editing of annotations is a client-side environment is contradicted by the clear disclosures in the ’286 Patent. *See* Section II above; ’286 Patent, 23:12-37 (describing this form-based approach is based on “the cue point editor resid[ing] on a *server* terminal”).

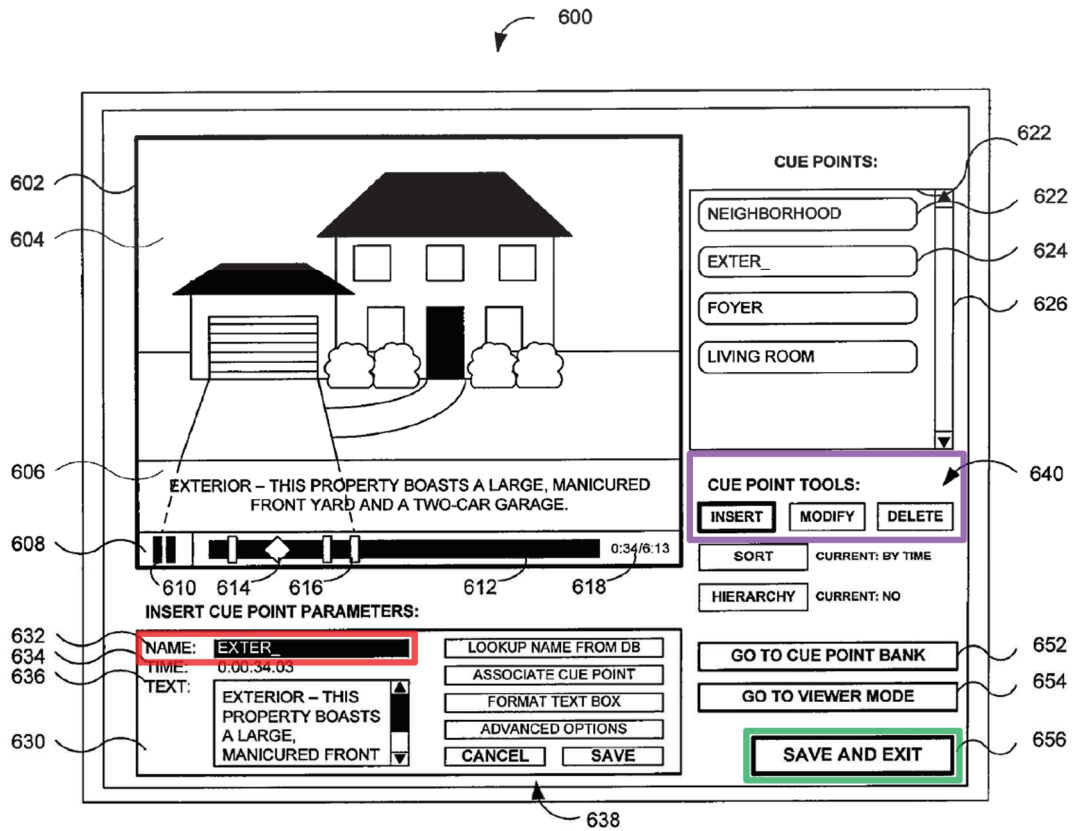
50. Gupta explicitly describes using a similar technique (ISAPI) to the CGI technique described in the ’286 Patent, for receiving client form data. Gupta, ¶[0047] (referencing ISAPI); EX1028 (ISAPI “provides the same functionality as CGI, but can be executed faster than CGI”); ’286 Patent, 18:15-17 (referencing CGI).

51. Thus, Gupta discloses server-side software for editing cue points that are created/modified at the server using cue point information and parameters received from a client.

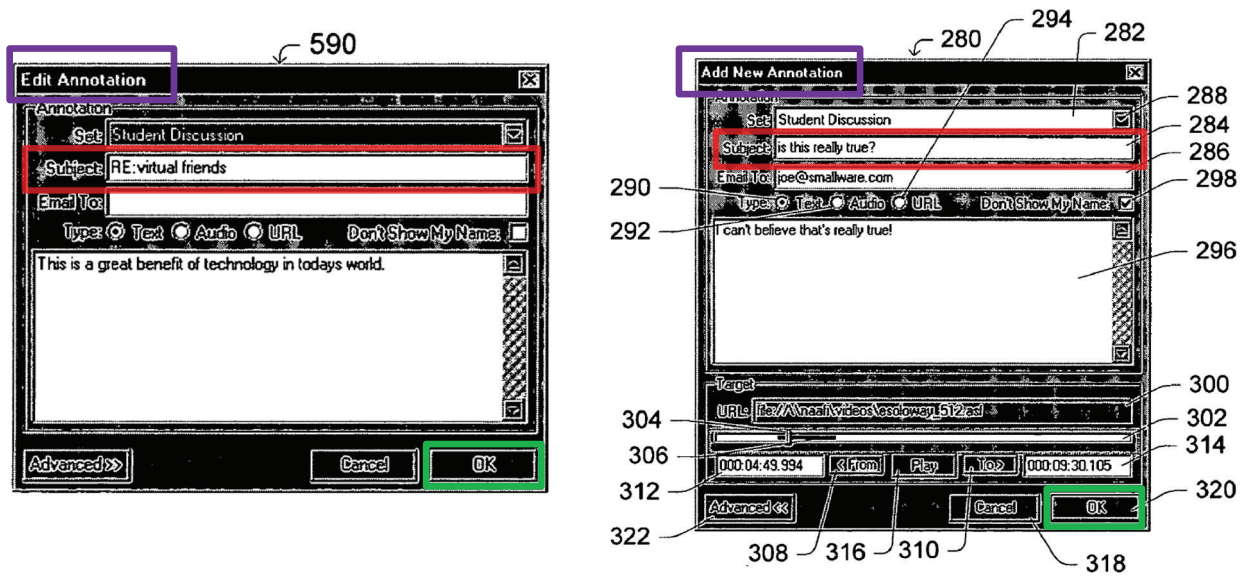
3. Gupta Discloses the Same Server-Side Methodology for “Creating”/“Modifying” a Cue Point as Described in the ’286 Patent

52. I also believe that Dr. Mahon’s characterization of Gupta editing/creating cue points at the client side rather than the server side, is flawed, given that Gupta’s approach to providing a client user interface to edit/modify annotations generally mirrors the server-side VEE example disclosed in the ’286 Patent, which I discussed above in Section II.A.

53. Both Gupta and the ’286 Patent describe similar user interfaces presented to a client user for inputting cue point information. I have reproduced exemplary figures from Gupta and from the ’286 Patent below, and added annotations with the same color pattern. In particular, a cue point text field has been annotated in red, a cue point tool control for adding or modifying cue point information has been annotated in purple, and a save/OK button to transmit the information to the server and exit from the interface screen has been annotated in green:



'286 Patent, Fig. 6 (annotated).



Gupta, Figs. 21, 7 (annotated).

54. As seen, the user interfaces between Gupta and the '286 Patent are highly similar, including in their presence of these annotated features.

55. As I discussed in my Original Declaration, a POSITA would have understood and recognized Gupta's disclosures involving its "add annotation" functionality (Fig. 7) to extend in the same way to its disclosures regarding its "edit annotation" functionality (Fig. 21), including the operations that result from when the "OK" button in each screenshot is selected. EX1002, ¶995. I note that Dr. Mahon likewise confirms that Gupta's "Paragraph 100 describ[ing] the operation of the 'OK' button for Figure 7 ... also applies to Figure 21." Mahon Decl., ¶115 (original bracketing omitted).

56. Furthermore, not only are counterpart features present in both Gupta and the '286 Patent, both describe the same process taking place upon selection of their respective save/OK buttons. In particular, both Gupta and the '286 Patent describe that selection of this button causes submission of the entered cue point information to the server side for database storage. Gupta, ¶¶[0100], [0048], [0051]; '286 Patent, 28:18-19, 27:55-57 (describing the save function); *see also* Section II above.

57. Furthermore, I also note that both Gupta and the '286 Patent describe displaying their respective interfaces via a client web browser. *See* Sections II, III.A.1-2 above; Gupta, ¶¶[0046]-[0047], Figs. 1, 3. As such, "creation" or "editing" of a cue point does not occur in Gupta simply by a user inputting cue point information

into Gupta's Fig. 21 web form, in the same way that such actions do not occur in the '286 Patent simply by a user inputting cue point information into the '286 Patent's Fig. 6 web form. Rather, the "creation"/"editing" occurs after such inputted information is sent to the server side, where a cue point is created or edited by updating the database/storage entry at the server side.

58. The '286 Patent discloses providing a user interface to a client browser for entering cue point information and transmitting the information back to the server to "create" cue points server-side, and Gupta discloses the same.

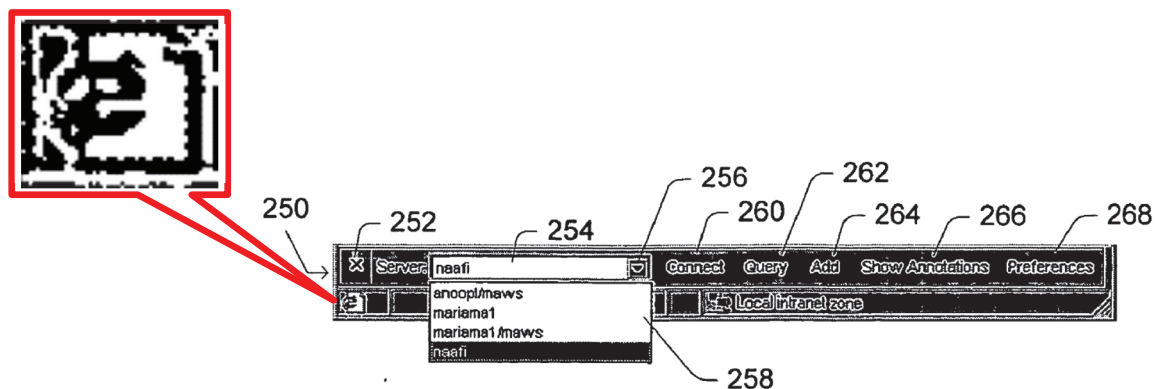
59. Dr. Mahon's opinion that "creation" of cue points occurs when a user inputs information into the forms at the client browser, rather than at the point the information is received at the server-side and stored in the server-side databases, is not credible. *E.g.*, EX2015, ¶¶99-108.

60. A POSITA would have had this understanding, especially since both Gupta and the '286 patent are focused on serving cue points to other viewers of the videos, which requires that such cue points be stored (and updated) at the server side. The '286 Patent's reference to "generating" a cue point at the server so as to save it, only further reinforces this understanding. '286 Patent, 23:12-37; *see also* Section II above.

B. Claim 15

61. Dr. Mahon opines that a user's selection of the "edit" tab in Gupta's Figure 15 is not an edit mode entry signal alleging that the transition from the screenshot of Figure 15 to the screenshot of Figure 21 is handled locally at the client side rather than via a signal sent of the network. Mahon Decl., ¶¶159-71. I disagree.

62. As I explained above, Gupta uses a web browser-based user interface, and specifically discloses web server functionality within annotation server 10 for this purpose. Gupta's screenshots being of a web browser is confirmed by Figure 6, which confirms these screenshots are from the Internet Explorer web browser:



Gupta, Fig. 6 (portion enlarged).

63. Therefore, Gupta's screenshots involve web access via a web browser. And as I discussed above, Gupta explicitly discloses web server functionality at the server side, including the functionality to dynamically generate web pages.

64. As yet additional confirmation that Gupta’s screenshots involve server-side web page generation, Gupta describes that its Fig. 21 interface “display[s] the **current** information for the annotation.” Gupta, ¶¶[0144] (emphasis added). Such “current” annotation information is maintained in server-side store 17 and retrieved by annotation server 10. Gupta, ¶¶[0032], [0048]; Mahon Dep., 152:21-157:19. Therefore, a POSITA would have understood that the Fig. 21 interface could not be generated locally since it is described as having *current* information, which information can only come from the server side where the current annotation information is located.

C. Claim 18

65. Dr. Mahon opines that that Gupta’s teaching of “read/write access rights (or associated credentials)” do not satisfy authentication information being received “over the network” by Gupta’s annotation server 10. Mahon Decl., ¶¶172-77. I disagree.

66. Gupta discloses “**annotation server 10** ... support[ing] 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.” Gupta, ¶[0090] ((9ea). Since Gupta’s annotation server 10 “controls the storage of annotations **and their provision to client computers**” (Gupta, ¶[0032] (emphasis added)), a POSITA would have understood that (i) the annotation creator must supply authentication information

(identifying users' read/write access information) to annotation server 10 so annotation server 10 can effect such controlled access to the creator's annotations, and (ii) a viewer user must also supply authentication information (*e.g.*, user/login credentials) for the annotation server 10 to provision to it access-restricted annotations. In fact, my Original Declaration explains this principle. EX1002, ¶¶131-33. Since annotation server 10 is located server-side and the creators and viewers are located client-side, Gupta discloses authentication information being received over the public network.

IV. Ground 2 (Fegan)

A. Claim 13

1. Fegan Discloses Server-Side Segmenting/Annotation Functionality

67. I understand that in its Institution Decision, the Board “invited” the parties “to address Fegan’s annotation and editing system, particularly the browser-based media player described in the above-quoted disclosures” and to also address “whether the segmenting and annotation tools of the browser-based media player are executed on the client side or server side of the network.” Institution Decision at 33.

68. Dr. Mahon opines that “all of Fegan’s disclosed embodiments carry out annotation editing operations locally through only client-side software” and that “Figures 5-7 of Fegan ... 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).” Mahon Decl., ¶183. Dr. Mahon also opines that Fegan “makes some vague high-level introductory statements” about its invention being maintained on the server side. Mahon Decl., ¶183. I disagree that Fegan’s segmenting/annotation functions are carried out by client-side software. I also disagree that Fegan’s references to server-side functionality are “vague,” “high-level,” and “introductory.”

69. Fegan makes numerous references to its system being “server-side,” being a “server-side environment,” being an “entirely server-side environment,” and being “maintained entirely on the server-side.” Such references are found across at least the Abstract, the Field of the Invention section, and the Summary of the Invention section.

The present invention relates to systems and methods for identifying, segmenting, collecting, annotating, and publishing multimedia materials. In some embodiments, the present invention provides a comprehensive server-side system for editing, annotating, managing and presenting multimedia events from a plurality of multimedia sources.

Fegan, Abstract (emphasis added).

FIELD OF THE INVENTION

The present invention relates to systems and methods for identifying, segmenting, collecting, annotating, and publishing multimedia materials. In some embodiments, the present

invention provides a comprehensive **server-side system** for editing, **annotating**, managing and presenting multimedia events from a plurality of multimedia sources.

Fegan, ¶[0003] (emphasis added).

SUMMARY OF THE INVENTION

The present invention relates to systems and methods for identifying, segmenting, collecting, annotating, and publishing multimedia materials that are stored on remote servers. In some embodiments, the present invention provides a comprehensive **server-side system** for managing and presenting multimedia events from a plurality of multimedia sources.

Fegan, ¶[0005] (emphasis added).

Thus, in addressing the limitations of the prior art, the present invention provides comprehensive systems for identifying, **segmenting**, collecting, **annotating**, and publishing multimedia materials. In preferred embodiments, the systems and methods are hosted in an **entirely server-side environment**, such that client computers can identify, segment, collect, annotate, organize, and publish complex multimedia presentations containing multimedia files or portions of multimedia files from a plurality of sources (e.g., any web site offering multimedia on the internet). In some such preferred embodiments, the identity of the files, the annotations, and the organization of the files is stored and managed on the host network. In preferred embodiments, the multimedia files are maintained only on the

third party web sites and are not copied onto the host or client computers. An advantage of such systems is the ability to manage complex presentations without burdening the client's computer hardware and allowing access to the presentation from any number of client computers. Specific illustrative, non-limiting embodiments of the present invention are described below in the Detailed Description of the Invention.

Fegan, ¶[0008] (emphasis added).

In preferred embodiments, the system is **maintained entirely on the server-side**. The server (host) system manages the appropriate databases (e.g., user information, multimedia file location information, annotations, start and stop points, etc.). The present invention is not limited to particular types or kinds of server-side software. In some preferred embodiments, the databases are driven by MYSQL, freely available database software for Unix and Win32/WinNT (see, e.g., MYSQL, Second Edition, by Paul DuBois, Pearson Education, 2003, herein incorporated by reference in its entirety). In some preferred embodiments, the features of the present invention are encoded in PHP and JavaScript, with an XML-based delivery and display (see, e.g., Advanced PHP Programming, George Schlossnagle, SAMS, 2003, herein incorporated by reference in its entirety). Software and tools for use in the present invention is described and available at MATRIX, Michigan State University (Lansing, Mich.)

Fegan, ¶[0010] (emphasis added).

Thus, in some embodiments, the present invention provides a system for manipulating, **annotating**, and managing a plurality of remote, multimedia files in a **server-side environment**, the system comprising a host computer network configured to carry out one or more of the following: a) identify multimedia files on an Internet web site, b) present the multimedia files to a client computer, c) receive playlist selection information (i.e., information pertaining to the identity of the multimedia files to be selected, annotated, or otherwise designated or manipulated) from the client computer, said playlist selection information comprising multimedia file identity and multimedia file start and stop points (i.e., designations of the starting and ending points of the multimedia file to be played in a presentation), d) **receive multimedia file annotation information from the client computer**, e) catalog (i.e., organize in a database or similar format) playlist selection information and multimedia file annotation information from a plurality of web sites selected by the client computer, and f) combine those multimedia files and annotations into online publications.

Fegan, ¶[0012] (emphasis added).

The present invention also provides a method for managing a plurality of multimedia files in a **server-side environment**, comprising one or more of the following steps: a) providing the systems of the present invention; b) identifying multimedia files on an Internet web site selected by the client computer; c) displaying identified multimedia files to the client computer; d)

receiving the playlist selection information from the client computer; e) **receiving** the multimedia file **annotation information from the client computer**; f) storing the playlist selection information and the multimedia file annotation information to generate a multimedia presentation; and g) providing the client computer access to the multimedia presentation.

Fegan, ¶[0015] (emphasis added).

70. In fact, Fegan makes a point to distinguish itself from prior client-side segmenting and annotating techniques which involved downloading the video to the client and using specialized client-side software to edit the video and specialized client-side authoring tools to publish the segmented video on the web. Fegan explains that, in contrast to these prior techniques, its “**technical tools**” “are **maintained on the server-side**, rather than forcing the user on the client-side to navigate the difficulties of the management of the multimedia files.”

Working with online, multimedia objects **has been** expensive and requires the user to have a good deal of technical expertise. For example, in order for a user to segment streaming media, the user would have to **download a copy** of the digital object and **use specialized software** to edit the digital objects into multiple files that contain the portion of content that the user desires. To publish the digital excerpts on the web, the user would then need--again--**specialized authoring tools and expertise** to construct

multimedia presentation using the clips that the user collects. **Downloading and editing** digital objects from the web not only presents **technical problems** for most users, but also brings with it problems of copyright and ownership.

The systems and methods of the present invention avoid these problems by, in preferred embodiments, only storing the url/uri (a web pointer) to the digital object. To isolate a piece of the digital object, the user only has to press "begin recording" and "end recording." This--again--does not store bits of the digital object, but simply the time offsets that the user has selected. The application then uses the url/uri and time offsets to play the portion of the digital object that the user has selected, and provides a means to add the user's own thinking to those portions of media. **The technical tools for carrying out the multimedia presentation are maintained on the server-side, rather than forcing the user on the client-side to navigate the difficulties of the management of the multimedia files.**

Fegan, ¶¶[0006]-[0007] (emphasis added).

71. I also note that the '286 Patent contains a strikingly similar criticism of client-side applications.

There remain, however, significant disadvantages to video marketing. One disadvantage is the cost of producing a quality result. Though most home computers today come with user-friendly photo editing software, much of the **video editing software and hardware is cumbersome**, expensive, and

difficult to learn. The result is that the average property seller or marketer must either spend a significant amount of money to hire a videographer to shoot and process video, or spend a significant amount of time learning to use a video editing system.

'286 Patent, 2:16-25 (emphasis added).

72. Dr. Mahon improperly downplays these teachings in Fegan and their parallels with the '286 Patent to miscast Fegan's tools for segmentation and annotation as client-side tools when Fegan clearly teaches they are used in an "entirely server-side environment." Mahon Decl., ¶¶180-202. I disagree with Dr. Mahon. In my opinion, a POSITA would have understood Fegan's teachings, including the repeated references to "server-side environment" (and "exclusively" "server-side environment") to indicate that the segmenting and annotating functionality is performed on the server side in Fegan. I also incorporate by reference the corresponding discussion in my Original Declaration. EX1002, Section XI.B.

2. Fegan's Segmenting/Annotation Tools are Distinct from the Media Player and are Server-side Functionality

73. In my opinion, the Board did not fully appreciate Fegan's teachings, as evidenced by the Board's Institution Decision inviting the parties "to address Fegan's annotation and editing system, particularly the browser-based media player

described in the above-quoted disclosures” and also address “whether the segmenting and annotation tools of the browser-based media player are executed on the client side or server side of the network.” Institution Decision at 33.

74. Fegan does not disclose segmenting/annotation tools “of the browser-based media player.” Rather, Fegan discloses that its “segmenting and annotation tools” are distinct and separate features from the media player. Unfortunately, Dr. Mahon does not correct the Board’s misunderstanding and instead continues to perpetuate this mischaracterization of Fegan. Mahon Decl., ¶¶181-84, 197-200, 203-06, 214, 221, 231.

75. Fegan discloses that the segmentation/annotation tools are not part of the media player, though both are displayed in a browser window. This arrangement is confirmed, for example, by Fegan’s description of the browser window including three different components: the media player, the segmentation controls, and the annotation controls (*e.g.*, text fields).

76. Fegan clearly discloses that the “browser window for editing of multimedia content” includes three distinct components—“[i] a compatible multimedia player, [ii] controls for generating time-based index points for virtual segmentation of the multimedia file, and [iii] “a means of entering annotative information,” *e.g.*, “text fields.” Fegan, ¶[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 [i] a multimedia file loaded into a compatible **multimedia player**, [ii] **controls** for generating time-based index points **for virtual segmentation** of the multimedia file, and [iii] 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, ¶[0080] (emphasis and bracketed enumerators added).

77. As yet further confirmation that both the segmenting and annotation features are separate from the multimedia player, Fegan characterizes the segmenting/annotation features as “additional tools,” separate from the media player.

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). ...

Fegan, ¶[0078] (emphasis added).

78. The segmenting/annotation tools are part of user interfaces that are loaded via web page (e.g., Figs. 4 and 7). Fegan is clear that these segmenting/annotation tools are server-side, as Fegan explicitly describes using a server-side web server which “responds to [client] request[s]” “by serving Web pages” to client browsers. If Fegan’s segmenting/annotation tools were client-side, they could be loaded exclusively by the client, and such web server access would not be needed to load the web page interfaces (e.g., Figs. 4 and 7) in the client browser.

As used herein, the term "client-server" refers to a model of interaction in a distributed system in which a program at one site **sends a request** to a program at another site and **waits for a response**. The requesting program is called the **"client,"** and the program that responds to the request is called the "server." In the context of the World Wide Web (discussed below), the **client is**

a "Web browser" (or simply "browser") that runs on a computer (e.g., desktop, cell phone, hand-held, etc.) of a user; the program which responds to browser requests by serving Web pages is commonly referred to as a "Web server."

Fegan, ¶[0037] (emphasis added); *see also* ¶¶[0043]-[0044].

79. Moreover, Fegan does not disclose the client running editing software for segmenting/annotating video—quite the opposite. And as I discussed above, Fegan distinguishes itself from such a client-centric approach. Fegan, ¶¶[0006]-[0007]. Instead, Fegan discloses that the client accesses the user's "personal 'portal' *hosted on the system*" to view and edit segments and annotations. Fegan, ¶¶[0009], [0081] (Figs. 5-7 depicting such portal access). Therefore, as I discussed above and as Fegan notes, such access to Fegan's personal portal web page "hosted on the system" is made via the client browser making requests to the web server for the personal portal webpage. Fegan, ¶[0037]. The data entry forms for entering or modifying the segment/annotation information on the personal portal are likewise requested by the client browser from the web server.

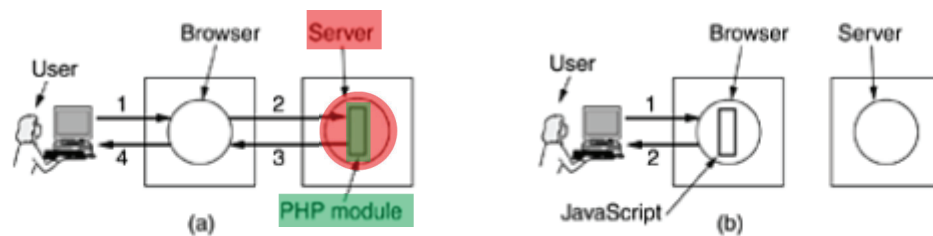
80. Fegan further discloses that its server side includes hardware/software server components "that serve the information content (*i.e.*, web site content) and "back end" hardware/software "to perform services for Web site users." Such server-side hardware/software would not be necessary if creating/modifying segments and annotations were performed client-side.

As used herein, the term "web site" refers to a computer system that serves informational content over a network using the standard protocols of the World Wide Web. Typically, a Web site corresponds to a particular Internet domain name and includes the content associated with a particular organization. As used herein, the term is generally intended to encompass both (i) the hardware/software server components that serve the informational content over the network, and (ii) the "back end" hardware/software components, including any non-standard or specialized components, that interact with the server components to perform services for Web site users.

Fegan, ¶[0042] (emphasis added).

81. Finally, Fegan also discloses the use of PHP scripts for creating and modifying segments/annotations (*see* Fegan, ¶[0010]), which as I discussed above in Section II.B (including with reference to Tanenbaum, 496-98), is server-side functionality used to provide information via web form information from a client browser to the server for processing and storage. Tanenbaum confirms that PHP is executed server-side:

Figure 7-37. (a) Server-side scripting with PHP. (b) Client-side scripting with JavaScript.



Tanenbaum, 500 (FIG. 7-37 (annotated in color)).

82. Therefore, based at least on these disclosures in Fegan and as I discussed above, a POSITA would have understood Fegan to teach the use of server-side processing to create/modify segments and annotations and provide dynamic web pages back to the client.

3. Fegan's Media Player Operates in the Same Manner as the '286 Patent's Sole Media Player Example

83. As I discussed above, Fegan describes its segmenting/annotation tools as being distinct from its media player, and therefore the media player is irrelevant when analyzing Fegan's segmenting/annotation tools. But regardless, Fegan's media player is arranged in the same way as the only media player example set forth in the '286 Patent, thereby further undercutting Dr. Mahon's opinion that Fegan does not satisfy the claims based on its media player.

84. Fegan discloses a browser-based multimedia player displayed in a browser window.

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. ...

Fegan, ¶[0076] (emphasis added).

After a desired multimedia file has been selected, 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] (emphasis added).

85. The '286 Patent discloses only one example of a video player, which mirrors Fegan's. The '286's Patent's Figure 6 user interface depicts video frame 602 in an interface window.

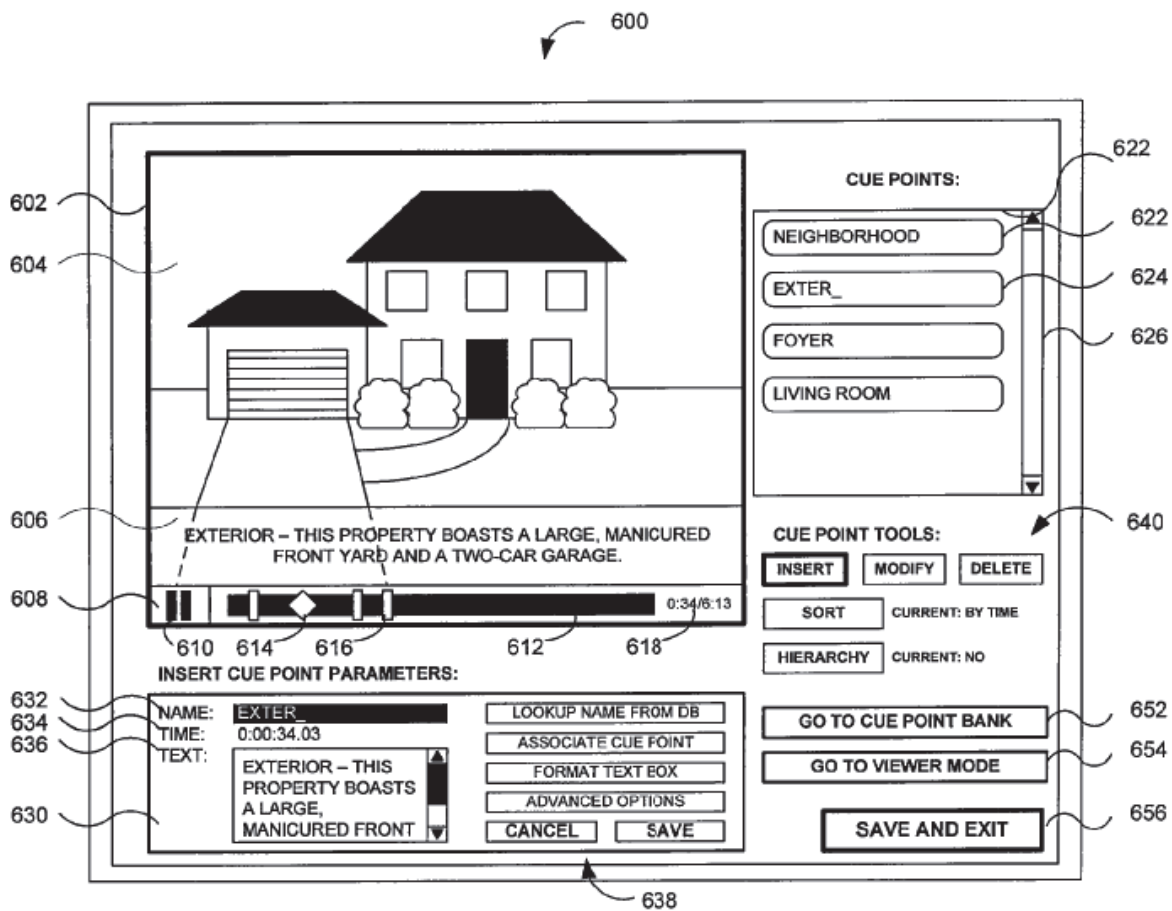


Fig. 6

'286 Patent, Fig. 6.

The cue point editor screen 600 comprises a video frame 602. The video frame 602 may be configured to display video data and to allow for interactive video manipulation. The video frame 602 may comprise a video footage area 604, a comment area 606, a navigation area 608, and any other useful areas.

'286 Patent, 26:6-11 (emphasis added).

86. The only example the '286 Patent provides for implementing that video player is as a “video player application (like Adobe® Flash®) embedded in the VEE GUI.” EX1001, 18:12-17. But Adobe Flash Player is **client**-side software, installed at the **client**. EX1032, 10.

87. Additionally, both Fegan and the '286 Patent disclose that video files are not maintained at the client, and instead are served by the server side. Fegan, ¶¶[0078], [0085]; '286 Patent, claim 1 (“stream ... the video information to the client terminal ... without storing the video file in its entirety to the client terminal”).

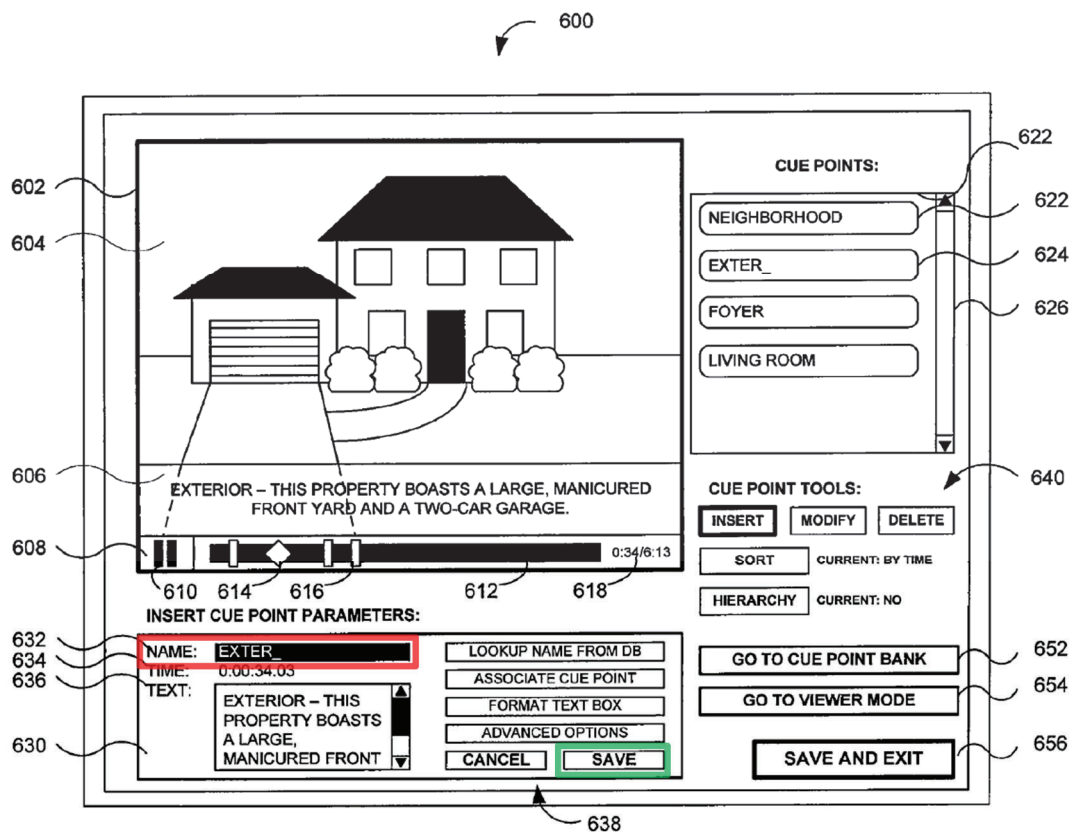
88. As the characteristics and operation of Fegan’s media player parallel those of the '286 Patent’s media player, I disagree that Fegan’s media player provides a reason to distinguish over the '286 Patent Challenged Claims.

4. Fegan Discloses the Same Server-Side Methodology for “Creating”/“Modifying” a Cue Point as Described in the '286 Patent

89. Dr. Mahon opines that in Fegan, client-side software is responsible for creating or modifying segments and annotations, not server-side software. Mahon Decl., ¶¶197-200. I disagree, at least because Fegan teaches the same general user interface and operation that is disclosed in the '286 Patent for server-side cue point creation/modification.

90. First, both Fegan and the '286 Patent describe web browser-based user interfaces displayed on a client device for user input of cue point parameters, and

both user interfaces are similar to each other. I have reproduced exemplary figures from each below. In each, I have annotated a cue point text field in **red** and a save/add/update button to save the changes in **green**:



'286 Patent, Fig. 6 (annotated); *see also id.*, 25:64-66.

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.

The screenshot shows a web browser window with the URL `http://kcar/media/sound_editor.php?soundfile=http://dlv1.matrix.msu...`. The interface includes a play button and a progress bar at the top. Below this, there are instructions and controls for recording a segment. The 'Start Position' is set to 4480 and the 'End Position' is set to 8860. There are buttons for 'Start Recording', 'End Recording', 'Full Sound', and 'Review Clip'. The 'Title' field contains the text 'flint sit down strike' and is highlighted with a red rectangle. The 'Annotation' field contains the text 'this is my annotation/thoughts on this sound'. The 'Folder' dropdown menu is set to 'labor' and the 'Add' button is highlighted with a green rectangle.

Press play to start recording.

Press start to start recording and end to stop recording.

Start Position: 4480 Start Recording

End Position: 8860 End Recording

Full Sound Review Clip

Title: flint sit down strike

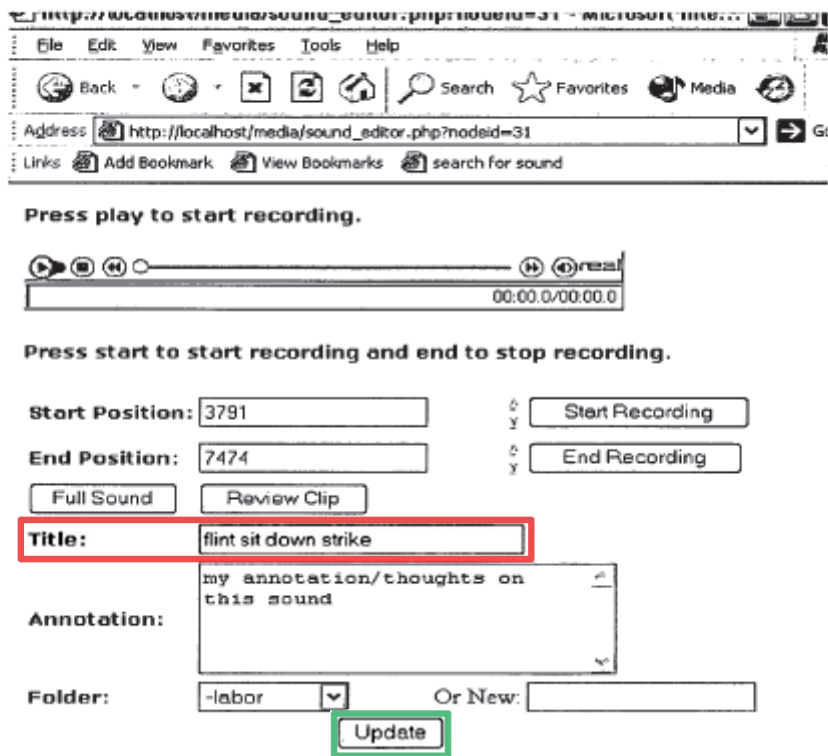
Annotation: this is my annotation/thoughts on this sound

Folder: labor Or New: labor Add

Fegan, Fig. 4 (annotated).

FIGURE 7

by clicking the edit button, a user can edit any of the information for that annotation including re-editing the time-offsets from the source.



Fegan, Fig. 7 (annotated).

91. Second, Fegan and the '286 Patent both describe the same action being performed in connection with selection of their respective update/OK buttons. In particular, both Fegan and the '286 Patent describe the resulting action being submitting the entered cue point parameters to the server side for storage. For example, Fegan discloses the use of PHP, while the '286 Patent describes the use of CGI. As I discussed above in Section II.B, both are server-side techniques for server-side processing and storage of information sent to the server via a client-side web browser form. Fegan, ¶¶[0015] (“server-side environment ... receiving ...

annotation information from the client”), [0012] (same), [0037] (client sends requests to web server), [0010] (PHP); ’286 Patent, 28:18-19, 27:55-57 (describing the save function).

92. Dr. Mahon opines that “[n]owhere does Fegan disclose that selecting the ‘Update’ button in Figure 7 causes the client to transmit a command to the server-side.” Mahon Decl., ¶221; *see also* ¶¶218-22. Dr. Mahon opines that “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.” Mahon Decl., ¶221. I disagree.

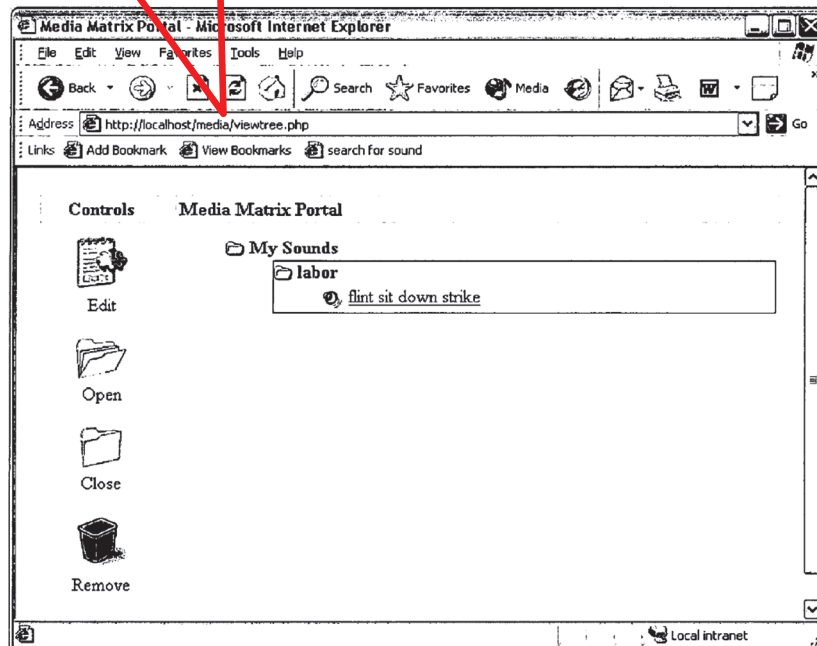
93. A POSITA would have understood that Fegan’s “Add” and “Update” buttons must send segments/annotation information to the server side for storage. Otherwise, Fegan’s system would have no way for the web server to present the current segments/annotations to the user via the user’s portal page (*e.g.*, Figs. 5-7). Such understanding is further based on Fegan’s figures showing a different server-side PHP script being executed depending on what action the client is requesting the server perform: (i) for the edit segment/annotation web page (Fig. 7), the client requests that the server execute the “sound_editor.php” server-side PHP script); and (ii) the personal portal web page (Fig. 5), the client requests that the server execute the “viewtree.php” server-side PHP script). The use of different PHP scripts further confirms that the server is dynamically generating webpages in response to client

requests. Since the user's portal page (accessed by the "viewtree.php" PHP script) presents "current" annotations in the Fig. 5 interface, any updates to annotations based on use of the Fig. 7 screenshot must have already been sent from the client to the server, *i.e.*, via a user selecting the "Update" button, for updating the server-side database. The current annotations would not be current on Fegan's Fig. 5 "portal" page if the "Update" button in the Fig. 7 edit screen only caused cue point information to be "locally updated on the client device," as Dr. Mahon opines.

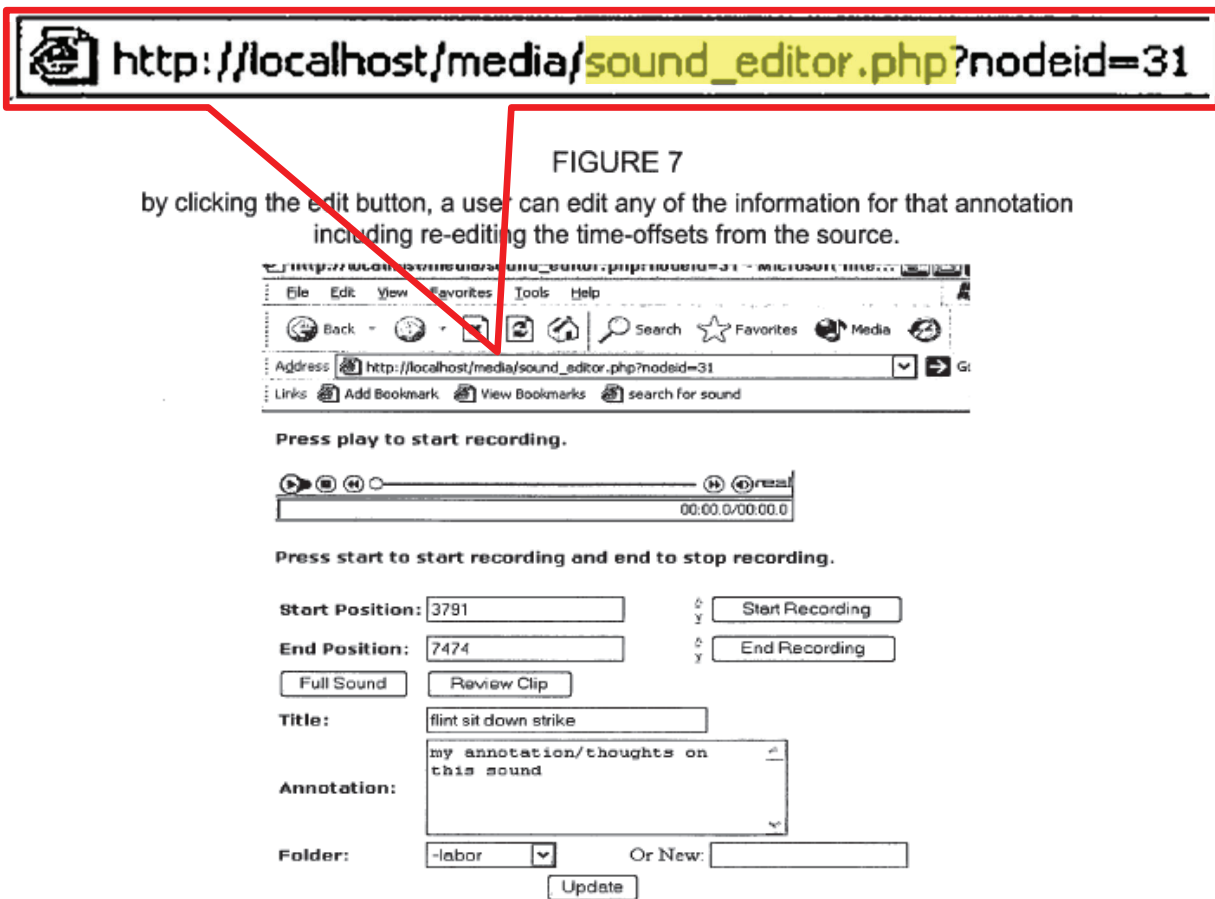


FIGURE 5

The folder created that appears on the user's portal page along with the annotation. 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, delete it, etc...)



Fegan, Fig. 5 (highlighted; portion enlarged).



Fegan, Fig. 7 (highlighted; portion enlarged); *see also id.*, ¶[0010]; Fig. 4 (the “Add” annotation screenshot likewise executing the “sound_editor.php” server-side PHP script).

... In some embodiments, each user has a **personal "portal" hosted on the system** of the present invention that **allows the user to edit** and organize **collected materials** and to create complex publications and presentations that include multimedia clips and annotations.

Fegan, ¶[0009] (emphasis added).

After a user designates a location for storage and retrieval of the annotative information, 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. ...

Fegan, ¶[0081] (emphasis added).

94. As I referenced in my Original Declaration, I been informed by counsel and understand that “a POSITA has ordinary creativity and is not an automaton.” EX1002, ¶34. Therefore, such a POSITA having ordinary creativity and accompanying common sense would have understood that user selection of the “Add” and “Update” buttons in Fegan causes the client to transmit a cue point editing command to the server with segment/annotation information for storage, thereby creating/modifying a cue point on the server.

95. I also note that such understanding likewise tracks the '286 Patent's reference to “generating” a cue point by saving it to the server, as I discussed above. *See* Section II.A above; '286 Patent, 23:12-37.

96. Third, both Fegan and the '286 Patent describe that their respective interfaces for sending information to the server for storage may be provided by the server and displayed on a client web browser. *See* Section II.A above (for '286 Patent); Fegan, ¶¶[0037], [0070], [0080]-[0081], Figs. 4-7.

97. The '286 Patent thus discloses its server-side VEE providing a user interface to a client browser to collect cue point parameters that are submitted to the server resulting in “creating” a cue point server-side. Fegan’s similar disclosure must be understood to also involve server-side functionality of providing a user interface to a client web browser to collect segment and annotation parameters and provide them to the server side so the server can “create” segments and annotations server-side.

5. Fegan Discloses Limitations [13.4]-[13.5]

98. Dr. Mahon opines that a POSITA would not have understood Fegan’s Fig. 7 “Update” button to send a cue point editing command to the server side. Fegan, ¶¶216-23. Dr. Mahon renders this opinion despite Fegan touting itself as providing an “entirely server-side environment” (Fegan, ¶¶[0008], [0012], [0015]) and despite Dr. Mahon conceding that Fegan stores and serves cue points server-side (Mahon Dep., 172:15-173:6).

99. I disagree with Dr. Mahon. In my opinion, Dr. Mahon’s opinions do not accurately reflect a POSITA’s understanding of Fegan’s Fig. 7.

100. Fegan explicitly discloses a “server-side environment” that “receiv[es] the multimedia file annotation information from the client computer” and “stor[es] ... the multimedia file annotation information.”

The present invention also provides a method for managing a plurality of multimedia files in a server-side environment, comprising one or more of the following steps: a) providing the systems of the present invention; b) identifying multimedia files on an Internet web site selected by the client computer; c) displaying identified multimedia files to the client computer; d) receiving the playlist selection information from the client computer; e) receiving the multimedia file annotation information from the client computer; f) storing the playlist selection information and the multimedia file annotation information to generate a multimedia presentation; and g) providing the client computer access to the multimedia presentation.

Fegan, ¶[0015] (emphasis added).

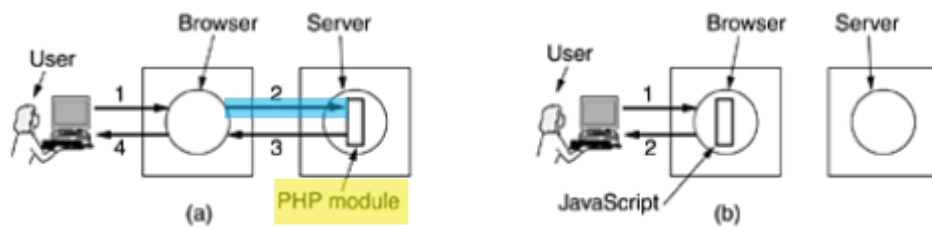
Thus, in some embodiments, the present invention provides a system for manipulating, annotating, and managing a plurality of remote, multimedia files in a server-side environment, the system comprising a host computer network configured to carry out one or more of the following: a) identify multimedia files on an Internet web site, b) present the multimedia files to a client computer, c) receive playlist selection information (i.e., information pertaining to the identity of the multimedia files to be selected, annotated, or otherwise designated or manipulated) from the client computer, said playlist selection information comprising multimedia file identity and multimedia file start and stop points (i.e., designations of the starting and ending points of

the multimedia file to be played in a presentation), d) **receive multimedia file annotation information from the client computer**, e) catalog (i.e., organize in a database or similar format) playlist selection information and multimedia file annotation information from a plurality of web sites selected by the client computer, and f) combine those multimedia files and annotations into online publications.

Fegan, ¶[0012] (emphasis added).

101. Fegan further emphasizes the use of web server (*i.e.*, server-side) functionality by identifying the use of PHP to implement its invention. As I discussed above in Section II.B, server-side web page generation (including both CGI and PHP) operates by the client sending a request—such request may be a request for an updated web page or may be a request to write/update an entry into a database at the server side. Tanenbaum confirms such requests being sent to the server for both CGI and PHP server-side techniques. I already discussed the CGI technique above in Section II.B (with reference to Tanenbaum), while Tanenbaum also explains the PHP technique, reproduced below:

Figure 7-37. (a) Server-side scripting with **PHP. (b) Client-side scripting with *JavaScript*.**



Tanenbaum, 500 (FIG. 7-37 (client-to-server command annotated in [blue](#))).

102. Dr. Mahon also confirmed during his deposition that such client-server messages are exchanged “[a]nytime you provide a service” under a client-server model. Mahon Dep., 73:18-21, 75:8-21 (“That’s how all client server architectures are going to work”), 77:12-19. And Dr. Mahon also confirmed this is the case for the HTTP protocol used in Fegan for communications between the client and the server. Mahon Dep., 77:12-19; Mahon Decl., ¶¶131-33; Fegan, ¶¶[0041], [0044].

103. Given Fegan’s disclosure that the server-side environment receives the annotation information from the client computer and stores it, a POSITA would have understood that a user selecting the “Update” button in Figure 7 is an example of causing an “Update” command (a cue point editing command) being sent from the client to the server over the network for updating an annotation stored server-side.

104. Given that Fegan specifically discloses “stor[ing] ... the multimedia file annotation information” after “receiv[ing] [it] from the client computer” (Fegan,

¶¶[0015], [0012]), a POSITA would have understood that when modifying an existing annotation (*e.g.*, Fegan’s Fig. 7 interface), the server’s “storing” operation modifies the corresponding existing annotation entry in server-side storage. Fegan, ¶¶[0005], [0008]. I understand that even Dr. Mahon concurs with such understanding. Mahon Dep., 172:15-173:6.

B. Claim 15

105. Dr. Mahon opines that Fegan’s disclosure of a user selecting the “Edit” button does not cause an edit mode entry signal to be sent from the client to the server “over the network.” Mahon Decl., ¶¶224-32. Specifically, Dr. Mahon opines that “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.” *Id.*, ¶228. I disagree.

106. Dr. Mahon ignores Fegan’s explicit teachings and the basic principles of web server operations—which would have been known to a POSITA. Indeed, Fegan even explains these basic principles of web site interaction—a client browser “**sends a request** to a [web server] and waits for a response” and the web server “responds to browser requests **by serving Web pages.**”

As used herein, the term “client-server” refers to a model of interaction in a distributed system in which a program at one site **sends a request** to a program at another site and **waits for a response**. The requesting program is called the **“client,”** and the

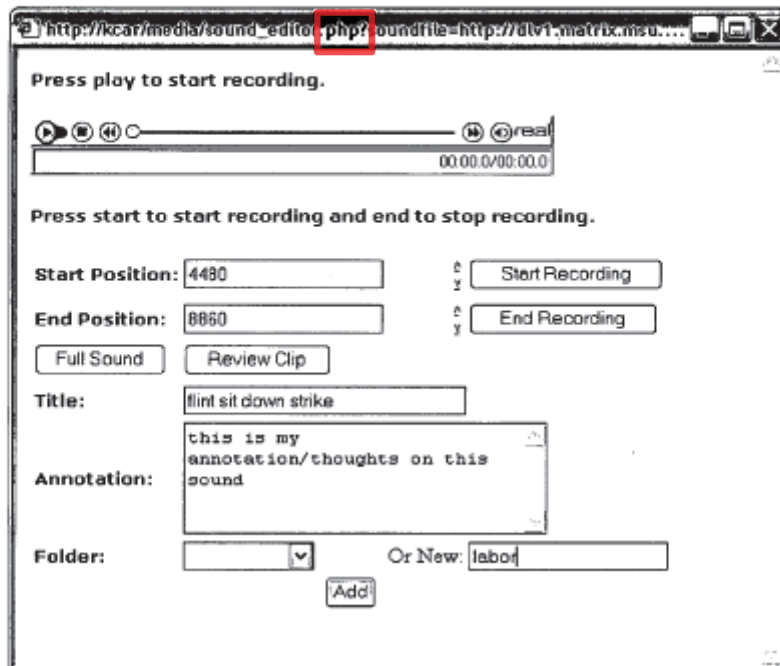
program that responds to the request is called the "server." In the context of the World Wide Web (discussed below), the client is a "Web browser" (or simply "browser") that runs on a computer (e.g., desktop, cell phone, hand-held, etc.) of a user; the program which responds to browser requests by serving Web pages is commonly referred to as a "Web server."

Fegan, ¶[0037] (emphasis added).

107. Moreover, Fegan discloses using PHP (*i.e.*, server-side functionality), which is executed at the server side to generate dynamic web pages that are returned to the client:

FIGURE 4

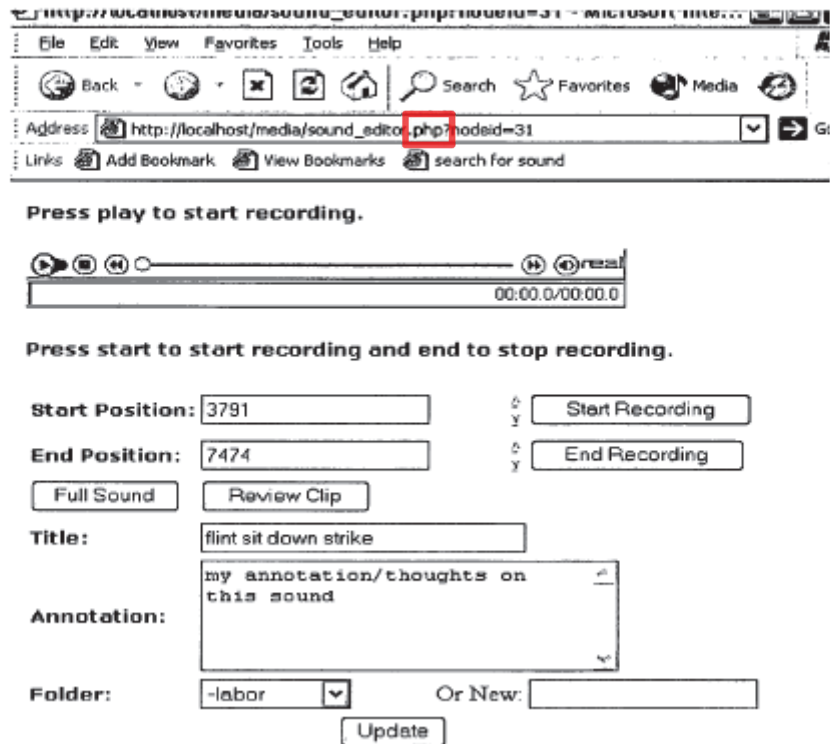
The system will load the sound into the editor and allow you to create a segment by pressing start and end recording and then adding a title, annotation, and dictating what folder this annotation should go in on your portal page.



Fegan, Fig. 4 (annotated).

FIGURE 7

by clicking the edit button, a user can edit any of the information for that annotation including re-editing the time-offsets from the source.



Fegan, Fig. 7 (annotated); *see also id.*, ¶[0010].

108. Notably as seen in these figures, Fegan specifically illustrates that the browser transition from Figure 5 to Figure 7 involves requesting a **different** PHP script (“viewtree.php” -> “sound_editor.php”) to be executed. Given that these two screenshots are based on different PHP scripts, a POSITA would have understood that each screenshot is separately retrieved from the web server and that selecting the “Edit” button in Figure 5 involves a client request (*i.e.*, the claimed “edit mode entry signal”) to the web server to execute the “sound_editor.php” PHP script and

serve the dynamic web page of Figure 7 to the client. I also note that while these screenshots refer to “localhost”, a POSITA would have clearly recognized that the use of localhost is merely for ease of generating the actual screenshots used for the patent application, particularly since Fegan’s main objective relates to a server-side environment distinct from the client. Fegan, ¶¶[0006]-[0008].

109. Therefore, in my opinion, Fegan clearly discloses claim 15.

C. Claim 18

110. Dr. Mahon further opines that, despite disclosing individual user portals and the use of security protocols, Fegan does not disclose that the server side receives “over the network, authentication information related to access authority” of the client terminal. Mahon Decl., ¶¶223-36. Again, I disagree.

111. Dr. Mahon did not attempt to rebut my testimony in my Original Declaration that a POSITA would have understood Fegan’s reference to a user’s *personal* portal page to involve sending authentication information to the server, to grant/deny such portal access (*i.e.*, the user must identify itself to the server by sending authentication information so the server can know which personal portal page the user can access to edit and organize their materials to create their presentations). EX1002, Section XI.E.

112. Therefore, in my opinion, Fegan clearly discloses claim 18.

V. Ground 5 (Vasilevsky in view of Logan)

A. Claim 13

1. A POSITA Would Have Been Motivated to Modify Vasilevsky's HAN to be a Public Network Such as Logan's Internet

113. Dr. Mahon opines that modifying Vasilevsky to be hosted on the public Internet in place of the Home Area Network would not work, and also opines that there would be no reasonable likelihood of success in the proposed combination. Mahon Decl., ¶¶ 325-48. I disagree.

114. To start, WellcomeMat and Dr. Mahon do not accurately characterize my opinions set forth in my Original Declaration, as they assert that I incorrectly opined that “a Home Area Network [HAN] located in the family home is a *public network*.” POR, 7 (citing EX1002, ¶320) (emphasis in original), 57-58; Mahon Decl., ¶306.

115. As I opined in my Original Declaration, I identify Vasilevsky's HAN as the portion of Vasilevsky that a POSITA would recognize could be (and would have been obvious to have been) modified to be a public network, such as the Internet as disclosed by Logan. For instance, I opined in my Original Declaration that “to the extent Vasilevsky does not explicitly disclose its network (HAN) is a ‘public’ network ... [a] POSITA would have been motivated to modify Vasilevsky's

network ('HAN' Home Area Network) to be a public network such as the Internet as taught by Logan.” EX1002, ¶¶320-22.

116. Dr. Mahon opines that “[i]f 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” and “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.” Mahon Decl., ¶¶329-30. I disagree.

117. As I referenced in my Original Declaration, “I been informed by counsel and understand that a POSITA has ordinary creativity and is not an automaton.” EX1002, ¶34. In my opinion, Dr. Mahon’s criticism of the proposed modification of Vasilevsky is given from the viewpoint of such an automaton, rather than a person of ordinary creativity who possesses common sense.

118. In my opinion, Vasilevsky is a cable TV system and a POSITA would have had the common sense to recognize that Vasilevsky’s cable line connections would need to be maintained. The proposed modification is not simply a matter of eliminating connections at will without accommodating these cable line connections

under the modification. Such accommodation is assumed to maintain the purpose of Vasilevsky's system and would have been well within a POSITA's skill level and common sense understanding. A POSITA would have also recognized that since at least the mid-1990s, cable providers have provided Internet and TV services over the same line, which would provide integrated support for both cable TV and public network (*e.g.*, Internet) connections to Vasilevsky's system. For instance, the Data Over Cable Service Interface Specification ("DOCSIS") is an international telecommunications standard that permits the addition of high-bandwidth data transfer to an existing cable television system, and has been employed since at least the late 1990's. *See* EX1030 (Wayback Machine archive of Wikipedia, DOCSIS). I am personally aware that DOCSIS has been (and to this day, continues to be) widely employed among cable TV providers as a protocol for providing Internet connectivity via cable lines. *See, e.g.*, EX1031.

119. Furthermore, as I explained in my Original Declaration, a POSITA would have recognized that Vasilevsky's system already has an existing Internet broadband connection and is therefore ready to implement the proposed modification for public network access. EX1002, ¶325. I also understand that Dr. Mahon concurs that Vasilevsky's system already has an Internet broadband connection. Mahon Dep., 178:7-179:1.

120. As I explained in my Original Declaration, Logan teaches the desirability of implementing a video streaming and bookmarking system over a public network, recognizing the benefits of providing such access to users outside the home to enjoy viewing, adding, or modifying video bookmarks. EX1002, ¶322. Petitioner's proposed modification simply expands Vasilevsky's system beyond the confined area of the home area network, by incorporating a public network (*e.g.*, Internet) to expand the same access to outside the home.

121. In my opinion, a POSITA would have had a reasonable expectation of success in the modification, because changing the HAN network used for transmitting data to a public network would have been straightforward given Vasilevsky's already-existing connections to the Internet and cable TV. As I discussed above, cable providers often provide TV and Internet access across the same cable, so the modification may be even as simple as re-configuring Vasilevsky's existing broadband connection to allow incoming connections and requests from clients elsewhere on the public network. This may be as simple as adjusting various network settings on a router or server in the system to allow access to servers on the home network by computers on the public Internet. I have personally made such modifications to my home router prior to the priority date of the '286 Patent and I found that instructions for making such modifications have been well-documented.

122. In either case, the modification would have been well within a POSITA's skill level and be simple common sense for an individual having an Electrical Engineering, Computer Engineering, or Computer Science background. And even if a POSITA elected to use an Internet service provider different from the cable TV provider, a POSITA would still have recognized the modification to be straightforward common sense. For example, it was not uncommon in households during the 1990's and early 2000's to have Internet access using DSL via a landline telephone service provider (*e.g.*, Bell Atlantic) and cable TV access using a cable provider (*e.g.*, Comcast). Setting up these connections would have been well within a POSITA's skill level.

123. Finally, I note that WellcomeMat's own evidence further confirms that a POSITA would have had such motivation to modify Vasilevsky. WO 2005/122025 ("Krikorian," assigned to Sling Media, Inc.), which pre-dates the '286 Patent, recognizes the benefits of enabling individuals to enjoy their TV content not only at home but outside the home as well, regardless of location and without dependence on physical cable connections.

While people spend a great deal of time watching television programming and other forms of audio-visual (A/V) content, they are also spending an increasing amount of time interfacing with computing devices such as personal computers, personal digital assistants, mobile phones, dedicated multimedia devices,

and other devices that, like the traditional television, include a display. These types of computing devices allow people to be increasingly mobile, but this mobility reduces the time people spend at home in front of their televisions. It would therefore be beneficial to enable people to enjoy their television programming and other forms of A/V content they now receive at home on these computing devices as well, regardless of location and without dependence on physical connections.

EX2032, ¶[0003] (emphasis added).

124. Krikorian also teaches another benefit of combining cable TV and Internet: enabling a user to access and control TV via a “remote network,” so the user can access his/her home TV system from distant locations such as an airport, work, school, hotel, café, etc.

This ability would enable several desirable applications. ... Another possible application would be to enable a user to access and control television and other A/V content from any number of remote networks where a broadband connection is available to the user (e. g., at an airport, at work, at school, in a hotel, in a cafe, at an acquaintance's house). Yet another application would be to enable a user to access and control television and other A/V content from a mobile phone or other computing devices that can be connected to a wide area network (e. g., GPRS, W-CDMA, CDMA-2000, 1XRTT,

1XEVDO, and the like). In various applications, **users are likely to want to access their media content** stored on devices, such as personal computers and other devices having storage, **from remote networks**. Nevertheless, network bandwidth and other limitations have made it difficult to provide an effective and enjoyable remote media experience for the user.

EX2032, ¶[0004] (emphasis added); *see also* ¶[0007].

125. Krikorian therefore further confirms the recognized desirability of expanding access of a home TV recording system to remote locations via a broadband connection, as I discussed in my Original Declaration. EX1002, ¶¶332-36. Ultimately, a POSITA would have recognized the proposed modification to Vasilevsky, based on Logan's teachings, as beneficial and would have had a reasonable expectation of success in making such a modification. Around 2002, I installed a music server on my home network that streamed music to a device connected to my stereo. The server also comprised a web server, and I configured my home router to allow connections to the server from the public Internet, so that I could access and play my music from my work place. The tasks I undertook in order to modify the router configuration were well documented and a POSITA would have been able to make such a configuration change.

2. Vasilevsky's Server Components Satisfy the Claimed VEE

126. Dr. Mahon opines that Vasilevsky's server side does not satisfy the claimed VEE because it is not a "software application." Mahon Decl., ¶¶307-24. He opines that Vasilevsky's server-side components "could have its functionality implemented as hardware (e.g., hardware including gates for logical operations) as opposed to being implemented using software." Mahon Decl., ¶320. I disagree that Vasilevsky's server components do not satisfy the claimed VEE.

127. First, I am not aware of any requirement that the claimed "video editing environment" is limited to software only and excludes hardware implementations. I understand that Dr. Mahon also testified that while the VEE would "typically" be software, it is "not restricted to software" and could "include[] hardware components."

Q Is the VEE software?

A It would typically be deployed as software.

Q Does that mean the VEE is a software application, then?

A It's not restricted to be software, but it can certainly be software, and it would be an application, in that instance.

If it included hardware components, then it would also include software. So it would have an application running on that hardware.

Mahon Dep., 79:3-13.

128. On this point, I agree with Dr. Mahon's deposition testimony, that the claimed VEE is not limited to software only and may include hardware components.

129. But I disagree with Dr. Mahon's opinions that a POSITA would have understood that Vasilevsky's server components could be implemented solely as hardware (or as Dr. Mahon refers to it, "hardware including gates for logical operations"). Mahon Decl., ¶320. Dr. Mahon cites to no evidence supporting this theory. In fact, I understand that Dr. Mahon testified that "[t]here would be ... various types of software running on any instantiation where you're providing a server for clients to access." Mahon Dep., 57:15-17 (emphasis added). In this regard, I agree with Dr. Mahon's testimony that servers run software. As Vasilevsky's VEE components are servers for client access, they would therefore likewise also be software-based.

130. Moreover, a POSITA would have recognized that such server components at the time would generally be software-based and would have recognized the many benefits of including software such as the ability to make bug fixes or improve any functionality via software updates. Further, a POSITA would have recognized that Vasilevsky's server components, including its application unit 134 which is "responsible for generating a graphical user interface (GUI) containing an electronic program guide (EPG) for display" (Vasilevsky, 15:17-20), are software-based.

B. Claim 15

131. Dr. Mahon opines that Vasilevsky does not disclose an edit mode entry signal transmitted over a network. Mahon Decl., ¶¶378-89. I disagree.

132. Vasilevsky describes that Application Unit 134 (within the centralized server) receives and converts user commands from clients to commands compatible with the media server. Vasilevsky, 15:15-17; EX1002, ¶338. And the communication between clients and the centralized server in Vasilevsky is over the network. Vasilevsky, Fig. 1, 11:9-13, 16:6-12; EX1002, ¶320.

133. Therefore, and as I explained in my Original Declaration, a POSITA would have understood that when a user selects the interface of Vasilevsky's Fig. 6 for a selected annotation, the centralized media server receives an edit mode entry signal (and does so over the network)—an indication to the computer system to enter an edit mode. EX1002, ¶364; *see also* ¶¶359-63.

C. Claim 18

134. Dr. Mahon opines that Vasilevsky does not disclose receiving authentication information over a network. Mahon Decl., ¶¶391-93. I disagree.

135. To start, Dr. Mahon offers no credible explanation as to how the server would only enable a user who created a bookmark to modify it without receiving some form of authentication information over the network from the user. Instead, he merely makes vague reference to a purported possibility that such authentication

information could be received by the server side “by direct input and other manners other than over the network.” It is unclear what Dr. Mahon’s reference to “direct input” means here, nor does Vasilevsky suggest that a client user has “direct input” access to the centralized server. Dr. Mahon also does not explain what “other manners other than over the network” are possible in Vasilevsky’s system.

136. Ultimately, Vasilevsky does not disclose any other connection between clients and the centralized server other than a network. Additionally, Vasilevsky’s figures depict usernames (*e.g.* “Schmidt”) attached to bookmarks, indicating that some authentication information must have been provided to the server in order for the client to be logged in with that username. Vasilevsky, Figs. 5-6.

137. Therefore, and as I explained in my Original Declaration, Vasilevsky’s system discloses the media server receiving authentication information from the client device over the network to permit identifying the user in question. This is necessary so the media server has the necessary information to determine whether this particular user has sufficient access for a particular bookmark. EX1002, ¶¶365-68.

VI. Conclusion

138. For the foregoing reasons, it is my opinion that claims 13-15 and 17-20 of the ’286 Patent recite subject matter that is anticipated and/or would have been obvious to a POSITA.

139. All statements made herein of my own knowledge are true, and all statements made on information and belief are believed to be true. Further, I am aware that these statements are made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001, and may jeopardize the validity of the application or any patent issuing thereon. I declare under penalty of perjury that the foregoing is true and correct.

A handwritten signature in black ink, reading "Henry H. Houh." The signature is written in a cursive, slightly slanted style.

Henry H. Houh, Ph.D.

Executed on April 7, 2025