

Petition for *Inter Partes* Review of  
U.S. Patent No. 6,415,316

UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE PATENT TRIAL AND APPEAL BOARD

---

Facebook, Inc.  
Petitioner

v.

Rembrandt Social Media, L.P.  
Patent Owner

U.S. Patent No. 6,415,316  
Filing Date: September 1, 1998  
Issue Date: July 2, 2002

Title: METHOD AND APPARATUS FOR IMPLEMENTING A WEB PAGE DIARY

---

**PETITION FOR *INTER PARTES* REVIEW  
OF U.S. PATENT NO. 6,415,316**

*Inter Partes* Review No. 2014-\_\_\_\_

## Table of Contents

	Page
I. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8(A)(1) .....	1
A. Real Party-In-Interest under 37 C.F.R. § 42.8(b)(1).....	1
B. Related Matters under 37 C.F.R. § 42.8(b)(2).....	1
C. Lead and Back-Up Counsel under 37 C.F.R. § 42.8(b)(3) .....	1
D. Service Information .....	2
E. Power of Attorney .....	2
II. PAYMENT OF FEES - 37 C.F.R. § 42.103 .....	3
III. REQUIREMENTS FOR INTER PARTES REVIEW UNDER 37 C.F.R. §§ 42.104 AND 42.108.....	3
A. Grounds for Standing under 37 C.F.R. § 42.104(a) .....	3
B. Identification of Challenge Under 37 C.F.R. § 42.104(b) and Statement of Precise Relief Requested .....	3
C. Requirements for Inter Partes Review 37 C.F.R. § 42.108(c).....	4
IV. TECHNOLOGY BACKGROUND RELEVANT TO THE '316 PATENT.....	5
V. DESCRIPTION OF THE CLAIMED SUBJECT MATTER .....	11
VI. CLAIM CONSTRUCTION UNDER 37 C.F.R. § 42.104(B)(3).....	17
A. “user system” .....	18
B. “diary program” .....	19
C. “content data” and “page design” .....	19
D. “cohesive diary page” .....	21
E. “configuration” and “privacy level” information .....	22
VII. CLAIMS 1, 4, 17, 18, 20 AND 26 OF THE '316 PATENT ARE UNPATENTABLE .....	24
A. Ground 1 – Claims 1, 4, 17, 18 and 26 are Obvious Over Salas In View of Tittel (1997) and Parker .....	24
1. Claim 1 is Obvious Over Salas In View of Tittel (1997) and Parker .....	25

**Table of Contents**  
(continued)

	<b>Page</b>
a.    Claim 1[a] – “Diary Program” .....	26
b.    Claim 1[b] – “Diary Information” .....	31
c.    Claim 1[c] – Assembling the “Diary Page” .....	41
d.    Claim 1[d] – Change of “Diary Information” .....	47
e.    Claim 1[e] – Sending “New Diary Information” .....	49
2.    Claim 4 is Obvious Over Salas In View of Tittel (1997) and Parker .....	50
3.    Claims 17 and 26 Are Obvious Over Salas In View of Tittel (1997) and Parker for the Same Reasons as Claim 1 .....	52
4.    Claim 18 Is Obvious Over Salas In View of Tittel (1997) and Parker .....	55
B.    Ground 2 – Claim 20 Is Obvious Salas In View of Tittel (1997) and Parker, and Further in View of Angles .....	56
VIII. CONCLUSION .....	59

List of Exhibits

Exhibit No.	Title of Document
<b>1001</b>	U.S. Patent No. 6,415,316 to Joannes Van Der Meer
<b>1002</b>	Declaration of Edward R. (“Ed”) Tittel
<b>1003</b>	Rulings pertaining to claim construction in <i>Rembrandt Social Media, L.P. v. Facebook, Inc.</i> , No. 13-CV-00158 TSE (E.D. Va.)
<b>1004</b>	Transcript of Court Hearing dated June 6, 2013 in <i>Rembrandt Social Media, L.P. v. Facebook, Inc.</i> , No. 13-CV-00158 TSE (E.D. Va.)
<b>1005</b>	U.S. Patent No. 6,233,600 to Pito Salas et al.
<b>1006</b>	Ed Tittel et al., <i>More HTML for Dummies</i> (2d ed 1997), pp. xv-xxv, 57-84, 153-180, 341-364
<b>1007</b>	Ed Tittel et al., <i>Foundations of World Wide Programming with HTML and CGI</i> (1995), pp. xxi-xxxvi, 125-144
<b>1008</b>	William Martiner, <i>Visual Basic Programmer’s Guide to Web Development</i> (1997), pp. vii-xiii, 47-97
<b>1009</b>	David Fox et al., <i>Web Publisher’s Construction Kit with HTML 3.2</i> (1996), pp. vii-xxv, 5-47, 51-61, 629-655
<b>1010</b>	M. Frans Kaashoek et al., <i>Dynamic Documents: Mobile Wireless Access to the WWW</i> (1995)
<b>1011</b>	U.S. Patent No. 5,729,734 to Robert D. Parker et al.
<b>1012</b>	U.S. Patent No. 5,933,811 to Paul Angles et al.
<b>1013</b>	U.S. Patent No. 6,289,362 to Joannes Van Der Meer

Petition for *Inter Partes* Review of  
U.S. Patent No. 6,415,316

Petitioner Facebook, Inc. (“Petitioner”) respectfully submits this Petition for *Inter Partes* Review under 35 U.S.C. §§ 311-319 and 37 C.F.R. § 42 of claims 1, 4, 17, 18, 20 and 26 of U.S. Patent No. 6,415,316 [Ex. 1001] (“the ’316 patent”).

**I. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8(a)(1)**

**A. Real Party-In-Interest under 37 C.F.R. § 42.8(b)(1)**

Petitioner, Facebook, Inc. is the real party-in-interest.

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

The Petitioner is aware of one pending litigation involving the ’316 patent: *Rembrandt Social Media, L.P. v. Facebook, Inc.*, Case No. 13-CV-00158 TSE (E.D. Va. filed February 4, 2013), pending in the U.S. District Court for the Eastern District of Virginia. The Petitioner was served on February 6, 2013.

The patent owner alleges that the Petitioner infringes claims 4, 20 and 26 of the ’316 patent. The Petitioner has denied infringement and contends that the asserted claims are invalid. Claim construction proceedings have taken place, as further explained in **Part VI** below. The litigation is currently stayed pending an appeal to the U.S. Court of Appeals for the Federal Circuit following the district court’s order excluding the patent owner’s damages expert.

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

Petitioner provides the following designation of counsel.

Petition for *Inter Partes* Review of  
U.S. Patent No. 6,415,316

LEAD COUNSEL	BACK-UP COUNSEL
Heidi L. Keefe (Reg. No. 40,673) hkeefe@cooley.com zpatdcdocketing@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Ave., NW, Suite 700 Washington, DC 20004 Tel: (650) 843-5001 Fax: (650) 849-7400	Mark R. Weinstein mweinstein@cooley.com zpatdcdocketing@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Ave., NW, Suite 700 Washington, DC 20004 Tel: (650) 843-5007 Fax: (650) 849-7400

Petitioner will file a motion to appear *pro hac vice* for proposed back-up counsel, Mark R. Weinstein, an experienced patent litigator and counsel for Petitioner. The earliest date on which such motion may be filed is twenty-one (21) days after service of this Petition on the patent owner. See “Order – Authorizing Motion for Pro Hac Vice Admission,” in IPR2013-00639.

**D. Service Information**

This Petition is being served by Federal Express to the attorney of record for the '316 patent, THOMAS | HORSTEMEYER, LLP, 400 Interstate North Parkway SE, Suite 1500, Atlanta, GA 30039. Facebook may be served at the address provided immediately above in Section I.C. The Petitioner also consents to electronic service by e-mail at the e-mail addresses provided above.

**E. Power of Attorney**

Filed concurrently with this Petition in accordance with 37 C.F.R. § 42.10(b).

**II. PAYMENT OF FEES - 37 C.F.R. § 42.103**

This Petition requests review of six claims of the '316 patent, therefore, no excess claim fees are required. A payment of \$23,000 is submitted herewith. 37 C.F.R. § 42.15. This Petition meets the fee requirements of 35 U.S.C. § 312(a)(1).

**III. REQUIREMENTS FOR *INTER PARTES* REVIEW UNDER 37 C.F.R. §§ 42.104 AND 42.108**

**A. Grounds for Standing under 37 C.F.R. § 42.104(a)**

The Petitioner certifies that the '316 patent is available for *inter partes* review, and that the Petitioner is not barred or otherwise estopped from requesting *inter partes* review on the grounds identified in the present Petition.

**B. Identification of Challenge Under 37 C.F.R. § 42.104(b) and Statement of Precise Relief Requested**

The Petitioner respectfully requests that the Board initiate *inter partes* review of claims 1, 4, 17, 18, 20 and 26, and find those claims unpatentable. This Petition cites the following prior art references:

<b>Exhibit No.</b>	<b>Title of Document</b>
<b>1005</b>	U.S. Patent No. 6,233,600 to Pito Salas et al. ("Salas")
<b>1006</b>	Ed Tittel et al., <i>More HTML for Dummies</i> (2d ed 1997), pp. xv-xxv, 57-84, 153-180, 341-364 ("Tittel (1997)")
<b>1011</b>	U.S. Patent No. 5,729,734 to Robert D. Parker et al. ("Parker")
<b>1012</b>	U.S. Patent No. 5,933,811 to Paul Angles et al. ("Angles")

Petition for *Inter Partes* Review of  
U.S. Patent No. 6,415,316

Each of the references listed above qualifies as prior art to the '316 patent under at least 35 U.S.C § 102(a), § 102(b) or § 102(e)(2) (pre-AIA). The grounds on which this Petition is based are listed in the table below.

Ground	Claims	Basis for Challenge
1	1, 4, 17, 18, 26	Unpatentable over Salas in view of Tittel (1997) and Parker under 35 U.S.C. § 103(a)
2	20	Unpatentable over Salas in view of Tittel (1997) and Parker, and further in view of Angles, under 35 U.S.C. § 103(a)

A detailed explanation of why each challenged claim is unpatentable under the statutory grounds identified above is provided in **Part VI** below.

This Petition and the Declaration of Edward R. Tittel (“Tittel Decl.”) [**Ex. 1002**], submitted herewith, cite additional prior art materials for purposes of providing a technology background and describing the state of the art at the time of the alleged invention. These materials are further discussed in the accompanying declaration from Mr. Tittel, an expert with more than three decades of experience in the computer software field.

**C. Requirements for *Inter Partes* Review 37 C.F.R. § 42.108(c)**

*Inter partes* review of claims 1, 4, 17, 18, 20 and 26 should be instituted because this Petition establishes a reasonable likelihood that the Petitioner will prevail with respect to at least one claim. See 35 U.S.C. § 314(a). Each limitation



of each challenged claim is disclosed and/or suggested by the prior art.

#### **IV. TECHNOLOGY BACKGROUND RELEVANT TO THE '316 PATENT**

The accompanying Tittel Declaration provides a technology background and describes the state of the art at the time of the alleged invention. (See Tittel Decl. ¶¶ 15-36.) This section will provide a summary of that description.

##### **A. The World Wide Web and HTML**

The “World Wide Web” (or “Web”) refers to a network of servers connected over the Internet that provide documents typically arranged in a format known as the HyperText Markup Language (HTML). HTML documents, commonly known as “web pages,” can be accessed using a “web browser” on the user’s computer using a protocol known as the Hypertext Transfer Protocol (HTTP). HTML documents may contain text, images, audio, video, links to other web pages or style sheets, and executable programs or scripts. (Tittel Decl. ¶ 16.)

The two most popular web browsers as of September 1998 were Internet Explorer (from Microsoft) and Navigator (from Netscape). A web browser issues requests to servers called “web servers,” which respond by sending information such as HTML documents to the browser. The browser and server, therefore, work together to deliver web pages and display them to a user. (*Id.* ¶ 17.)

Navigation through the Web with a web browser typically involves an address, known as a “Uniform Resource Locator” or “URL,” that specifies the location of a web page or other resource on the Internet. A URL is a string that follows a specific format (e.g., “http://www.uspto.gov/”) that identifies the communication protocol (e.g., “http”), the web server where the resource is located (“uspto.gov”), and may provide additional information such as a directory or name associated with the requested resource. (*Id.* ¶ 18.)

A web page may include various content items such as text, graphics, audio and video. The HTML code on that page tells the web browser generally how the page should appear and how to display the content for the page. The HTML code for a given web page may include information that designates the typeface and style for text, the format and structure of the page, and arrangement of content on the page. A more advanced technique, known as “style sheets,” provides even greater control over the appearance of a web page. A style sheet can be used, for example, to provide a set of formatting and display rules that can be applied across certain types of elements. (*Id.* ¶ 20.)

Style sheets can either be embedded within the HTML of a web page or contained in a separate file. In the latter scenario, commonly known as an “external style sheet,” the HTML file typically contained a tag known as a <LINK>

tag that identified the URL for the style sheet file. When the web browser encountered the <LINK> tag, it downloaded the style sheet file from the server and used it to properly display the web page. (*Id.* ¶ 21.)

Using external style sheets was exceedingly simple. Mr. Tittel's 1997 textbook regarding HTML stated: "When you have all of the pieces and parts of a style sheet, how exactly do you link style definitions to your Web pages? It's so simple it's scary." (*Id.* ¶ 22 (quoting Ed Tittel et al., *More HTML for Dummies* (2d ed 1997) [Ex. 1006], at 68).) Linking an external style sheet simply involved adding a single <LINK> tag to the HTML file specifying, among other things, the name of the style sheet file as stored on the web server:

```
<LINK REL=StyleSheet HREF="mystyle" TYPE="text/css" MEDIA=screen>.
```

(*Id.* ¶ 22.)

In the example above, the "HREF" attribute specifies the name and location of the style sheet file. In that example, the sheet named "mystyle" is located in the same directory on the server as the web page, so a fully-qualified URL is unnecessary. If this <LINK> tag were included in a home page located at <http://www.uspto.gov/home.html>, for example, the web browser would find the style sheet at <http://www.uspto.gov/mystyle>. (*Id.* ¶ 23.)

**B. Scripts, Applets and Other Browser-Based Computer Programs**

Although HTML provided a means for defining the layout and format of web-based content for display in a browser, it lacked the ability to make that content interactive. The industry responded by developing a variety of technologies to create and embed downloadable programs into web pages. Two well-known examples that existed prior to September 1998 included Java (from Sun Microsystems) and ActiveX (from Microsoft). For example, programs written in Java, which were typically called *applets*, could be used to create animation, play music, play video games, display real-time video, and perform many other interactive tasks. (*Id.* ¶¶ 26-27.) Java and ActiveX provided a straightforward way to embed an executable computer program in the HTML for a web page, which would cause the browser to download the program from the web server and execute it at the user's computer. (*Id.* ¶ 25.)

"ActiveX" was a similar and competing technology to Java. ActiveX programs were known as "ActiveX controls" or "ActiveX components," and they could be embedded in a web page using techniques exceedingly similar to Java. Suppose a developer wrote a calendar program as an ActiveX control called "Calendar.ocx" (ActiveX controls typically took an ".ocx" or ".cab" extension), and

wanted to insert that program into a web page. The following exemplary HTML could be added to embed that ActiveX control into a web page:

```
<OBJECT CLASSID="clsid:54DDB4E0-829D-11CE-AC36-00AA00B47DD3"  
CODEBASE="http://tittel.com/My_Calendar.ocx" HEIGHT=300  
WIDTH=200> </OBJECT>
```

When the web browser encounters the <OBJECT> tag above, it downloads the ActiveX control from the remote server if it has not already been downloaded, and executes it on the user's computer. The "CLASSID" attribute is typically a sequence of characters specifying specifies a unique identifier for the ActiveX control or component. The user's computer uses the "CLASSID" attribute to determine whether the ActiveX control or component has already been downloaded. The "CODEBASE" attribute contains a URL identifying the name and location of the ActiveX control. In this case, the ActiveX control would be located at <http://tittel.com/My\_Calendar.ocx> (*Id.* ¶ 29.)

If the web browser encounters the ActiveX component for the first time, "the browser must first download and install the component before it is used. For this reason, when you include an ActiveX component in a Web page, you must also specify the **CodeBase** (i.e., URL) from which the browser can download the control." (*Id.* ¶ 29 (quoting William Martiner, *Visual Basic Programmer's Guide to*

*Web Development* (1997) [Ex. 1008], p. 77-78) (bold in Martiner.) The ending </OBJECT> tag signals the end of the information in the <OBJECT> element. (*Id.*)

### C. Browser-Based Computer Programs That Generate Web Pages

It was also known prior to September 1998 that computer programs running locally on the user's computer (such as Java applets or ActiveX components) could generate HTML for a web page that would be displayed in the user's web browser. This technology was described in an article published in 1995, M. Frans Kaashoek et al., *Dynamic Documents: Mobile Wireless Access to the WWW* [Ex. 1010] ("Kaashoek"). Kaashoek describes a technique in which a web browser generates HTML-based web pages in the context of a mobile computing device. (Tittel Decl. ¶¶ 31-32 (describing Kaashoek).)

Another example of a system that relies on locally-executing software to generate a web page is disclosed in U.S. Patent No. 6,233,600 to Pito Salas et al. ("Salas") [Ex. 1005]. Salas describes a technique for creating an "eRoom," a web-based collaborative environment for organizing information and sharing it with other users. The system described in Salas, like that of Kaashoek, relies on software running in the user's web browser to generate eRoom web pages locally. (Salas, 2:20-27, 6:64-7:4.) Salas explains that this software in one embodiment can be implemented as an ActiveX control:

Generation, display and management of eRooms is controlled by a “page builder” application residing on the client workstation **12’**. The page builder application may be provided as specialized hardware or software operating on a general purpose machine. In some embodiments, the page builder application may be provided as an Active X control or a COM object.

(*Id.*, 6:57-63.) A detailed explanation of Salas is provided in Part **VII.A** below.

**V. DESCRIPTION OF THE CLAIMED SUBJECT MATTER**

**A. The Specification of the ’316 Patent**

The ’316 patent, which is entitled “Method and Apparatus for Implementing a Web Page Diary,” discloses a technique for creating and displaying a web page that stores web page links and other content. (’316 patent [Ex. 1001], 1:20-23.) The ’316 patent discloses a method and system for organizing and displaying information through a web page referred to in the patent as a “cohesive diary page.” As explained in the Summary of the Invention:

The present invention allows a user to create a “diary” containing multimedia references to Websites. These references (also called “content objects” or “objects”) can be addresses or URLs of, for example, text, bookmarks, images, programs, movies, etc. Many content objects are provided via the Websites of “content providers,” with the specific intent of making the content objects

available to a user to place in his diary. Other content objects can be copied from the diaries of other users. Still other content objects are entered by the diary owner himself.

(*Id.*, 2:2-11.) In one embodiment described in the specification, the diary page owner selects a visual “theme” for the pages of his or her diary, the theme “reflect[ing] how the diary owner wants to present himself and his diary to the world.” (*Id.*, 2:26-29.)

The specification provides a lengthy description of the diary creation system, but the claims address only one aspect of it – the mechanics of actually assembling and displaying the “cohesive diary page” in a user’s browser. This section of the Petition will therefore focus on the portions of the ’316 specification pertaining to this claimed process.

In order to assemble and display a “cohesive diary page” to the user, the server must send two pieces of information to the user’s system: (1) “**diary information**,” which described in more detail below, and (2) a “**diary program**” (or “**diary applet**” as it is referenced in the specification), which is an executable computer program that runs in the user’s web browser. The purpose of the “diary program,” as explained in more detail below, is to generate a diary page based on the “diary information” received from the server. (*Id.*, 7:1-3, 9:7-9.)



“Diary information” in the context of the ’316 patent comprises three types of information. First, the “diary information” includes the “content data” (or content objects as they are referenced in the specification) that may be displayed to a user. “A content object can be any type of object, including text, bookmarks, images, programs, movies, etc. The set of content objects inserted into the diary by the user is known as the ‘book content.’” (*Id.*, 5:21-25.)

Second, the “diary information” includes “page design” information, which is layout or style information that defines (at least in part) the visual appearance of a diary page. (*Id.*, 5:15-18 (“The page design defines the visual and audible appearance of the page. A page design provides slots for content entries or objects. The page design determines the size and location of these slots within the page.”).) The specification also uses the terms “presentation context” and “cover” to refer to this page design information. (*Id.*, 8:59-64, 17:60-61.) In one embodiment in the specification, the page design is represented as an HTML file that that provides a template for the diary page. (*Id.*, 17:60-64, Fig. 12.)

The third type of “diary information” is referred to in the specification as “configuration information,” which includes “privacy level information.” The patent explains that “configuration information” is used to determine whether a user is permitted to view or access content on a diary page. “If a user does not

have sufficient permission to view an object in a diary,” for example, “the diary may not make the object visible to the user, i.e., the user does not even know that the object exists, or it may present the object using an alternate representation.” (*Id.*, 5:63-67.) This is accomplished through use of “configuration information,” such as passwords, privacy levels or information identifying the authorized user. (*Id.*, 9:2-4 (“The configuration information, such as privacy level, passwords, or the full name of the user, is user-specific.”), 10:51-54 (“The passwords for a particular user’s diary pages are stored as configuration information in that user’s diary information **114**.”).)

After the “**diary program**” receives the three pieces of “**diary information**” from the diary server, the “diary program” generates HTML for the diary page based on the received diary information. (*Id.*, 7:1-3 (“Diary applet **112** reads diary information **114** received from the server and generates HTML **111** for one or more diary pages in accordance with diary information **114**.”); *see also id.*, 9:7-9.) The “diary program” can be implemented as a Java applet, an ActiveX control, or any other appropriate executable program. (*Id.*, 6:52-56.)

Generation of the diary page also takes into account the “**configuration information**” and “**privacy level information**” received from the server. As mentioned previously, this means that if a user does not have permission to view

an object in a diary, “the diary may not make the object visible to the user, i.e., the user does not even know that the object exists, or it may present the object using an alternate representation.” (*Id.*, 5:63-67; *see also id.*, 18:15-20, 17:19-21.)

**B. The Claims of the '316 Patent**

The two independent claims of the '316 patent addressed in this Petition – claims 1 and 17 – purport to recite a method and apparatus, respectively, for implementing the diary page generation process described in the preceding section. Claim 1 recites:

1. A method of organizing information for display, comprising:
  - [a] sending from a diary server to a user system, a diary program capable of being executed by a browser in the user system;
  - [b] sending diary information from the diary server to the user system, the information comprising content data including an associated time, a page design to specify the presentation of the content data, and configuration information for controlling behavior of a cohesive diary page, the configuration information including privacy level information;
  - [c] assembling the cohesive diary page by dynamically combining the content data and the page design in accordance with the configuration information for the cohesive diary page to be displayed by the diary program running in the browser;
  - [d] receiving by the diary server at least one request for at least one change concerning the diary information, from the diary program in the user system; and
  - [e] sending, by the diary server to the user system, new diary information for changing the cohesive diary page.

(’316, Claim 1.) The bracketed notations (*e.g.*, “[a],” “[b],” etc.) were added to facilitate easier identification of these limitations in this Petition. The other independent claim addressed in this Petition is claim 17. That claim is similar in many respects to claim 1 but was drafted as an apparatus claim:

17. An apparatus that displays and organizes information, comprising:
  - [a] a software portion configured to receive, by a user system from a diary server, a diary program capable of being run by a browser in the user system;
  - [b] a software portion configured to receive, by the user system from the diary server, diary information comprising content data including an associated time, a page design to specify the presentation of the content data, and configuration information for controlling behavior of a cohesive diary page, the configuration information including privacy level information;
  - [c] a software portion configured to assemble the cohesive diary page by dynamically combining the content data and the page design in accordance with the configuration information; and
  - [d] a software portion configured to display the cohesive diary page, by the diary program running in the browser.

(’316, Claim 17 (bracketed notations added for identification).)

Unlike claim 1, claim 17 does not include the additional steps from claim limitations 1[d] and 1[e] relating to changing diary information and receiving new diary information for the page. A similar requirement is addressed by claim 26:

26. The apparatus of claim **17** further comprising:
- a portion configured to receive, from the diary server, new diary information;
  - a portion configured to change content of the diary page without changing a general appearance of the diary page, in accordance with the new diary information.

(’316, claim 26.) The additional claims addressed in this Petition, claims 4, 18 and 20, are dependent claims that add nothing of patentable significance. This Petition will address those claims in more detail below.

**VI. CLAIM CONSTRUCTION UNDER 37 C.F.R. § 42.104(b)(3)**

A claim subject to *inter partes* review must be given its “broadest reasonable construction in light of the specification of the patent in which it appears.” 37 C.F.R. § 42.100(b). As the Federal Circuit has recognized, the “broadest reasonable construction” standard is different from the manner in which the scope of a claim is determined in litigation. *See In re Swanson*, 540 F.3d 1368, 1377-78, 88 U.S.P.Q.2d 1196, 1203 (Fed. Cir. 2008).

As noted in **Part I.B** above, the district court in the copending *Rembrandt Social Media, L.P. v. Facebook, Inc.* patent litigation has issued claim construction rulings with respect to certain terms of the ’316 patent that are recited in the challenged claims. A copy of the Court’s written rulings on claim construction is attached as **Exhibit 1003**. Attached as **Exhibit 1004** is a transcript of a June 6,

2013 conference in which the Court stated the basis for certain of its rulings.

Although the Board is not bound by claim construction rulings by the district court, the Petitioner respectfully requests that the Board adopt those constructions for at least the more significant terms, which are discussed below.<sup>1</sup>

**A. “user system”**

The term “**user system**,” which appears throughout independent claims 1 and 17, should be construed as “a computer that executes a browser for use by an end user.” The specification describes a “user system” as a conventional data processing system that includes at least a processor and a storage area such as a memory. (’316, 7:32-36; Fig. 2(a).) The storage area in the user system includes “a web browser **210** and diary information **114**. Browser **210** can be any appropriate browser, including but not limited to [Netscape] Navigator and [Microsoft

---

<sup>1</sup> Many of the terms addressed in the district court’s claim construction order pertain to terms that are not relevant to this Petition. This is because many of those terms are recited in claims of a different patent (U.S. Patent No. 6,289,362), or in claims of the ’316 patent that were withdrawn from the litigation by the patent owner. The discussion in the text, therefore, focuses on the more significant terms recited in the six claims challenged in this Petition.

Internet] Explorer.” (’316, 7:36-40.) This is consistent with the language of claims 1 and 17, which describe the “user system” as a computer system that runs a browser. (’316, Claim 1[a] (“sending from a diary server to a user system, a diary program capable of being executed by a browser in the user system...”).)

**B. “diary program”**

The term “**diary program**,” recited in independent claims 1 and 17 of the ’316 patent, should be construed as “a computer program for execution by the browser in the user system that generates a cohesive diary page.” The “diary program” is referenced in the specification as a “diary applet” and described as a Java applet, ActiveX control, or other executable program that executes in the browser in the user’s system. (’316, 6:52-56.) The purpose of the “diary program” is to generate the HTML for a diary page based on the received diary information. (’316, 7:1-3 (“Diary applet **112** reads diary information **114** received from the server and generates HTML **111** for one or more diary pages in accordance with diary information **114**.”); *see also* ’316, 9:7-9 (“In step **308** [of Fig 3], diary applet **112** generates one or more pages of the diary in HTML in accordance with the cover, content, and configuration information.”).)

**C. “content data” and “page design”**

The terms “**content data**” and “**page design**” are related in that they

represent two types of information used to generate the “cohesive diary page” recited in claims 1 and 17. (See, e.g., ’316, claim 1[c] (“assembling the cohesive diary page by dynamically combining the content data and the page design . . . for the cohesive diary page . . . .”); see also Tittel Decl. ¶¶ 43-46.) The specification describes “content data” (referred to in the specification as “content objects”) as information, such as text and images, that may be displayed on a diary page. (’316, 5:21-25 (“A content object can be any type of object, including text, bookmarks, images, programs, movies, etc. The set of content objects inserted into the diary by the user is known as the ‘book content.’”).) “Page design,” on the other hand, refers to layout or style information that defines the visual appearance of a diary page. (’316, 5:15-18 (“The page design defines the visual and audible appearance of the page. A page design provides slots for content entries or objects. The page design determines the size and location of these slots within the page.”).) The “content data” and the “page design” are “independent” of each other. (’316, 5:26-28 (“Unlike a traditional book, the book design and book content of a diary are independent. Both of [sic] the book design and the book content may be changed at any point in time.”).) The Petitioner therefore requests that the Board interpret these two terms as:

- “content data” – information that may be displayed to a user that is



independent of the page design; and

- “page design” – layout or style information that defines, at least in part, the visual appearance of a cohesive diary page, independent of the particular content of the page.

**D. “cohesive diary page”**

A “**cohesive diary page**,” as recited in claims 1 and 17, represents the result of the step of combining the “content data” and the “page design” information. (See, e.g., ’316, claim 1[c] (“assembling the cohesive diary page by dynamically combining the content data and the page design...”).) As explained in the specification, “[t]he diary software dynamically combines the diary’s book design and book content to present a cohesive view of the ‘book.’” (’316, 5:32-34.) The term “cohesive diary page,” consistent with the specification and claim language, should be understood as a diary page in which the content data and the page design are fully integrated for display.

The term “**diary page**,” a component of the larger term “**cohesive diary page**” discussed above, is simply a page containing content sent by a server to a user system. Although people often think of the word “diary” as referring to a personal and private journal viewed only by a single individual – the owner of the diary – who authors all content in the diary, the ’316 patent uses the term in a

broader way. (Tittel Decl. ¶ 40.) The content of a diary page may be inserted by the diary owner or by any other user who has permission to do so. ('316, 5:19, 6:9-11.) Additionally, “[t]he diary can be read concurrently by multiple users in different locations.” ('316, 5:48-49; Tittel Decl. ¶¶ 40-41.) A **“cohesive diary page”** under its broadest reasonable construction, therefore, should be construed as “a page sent by a server to a user system in which the content data and the page design are fully integrated for display.” (Tittel Decl. ¶¶ 40-41.)

E. **“configuration” and “privacy level” information**

**“Configuration information”** and **“privacy level information”** are two closely-related terms that appear adjacent to each other in claims 1 and 17. ('316, *e.g.*, Claim 1: “sending diary information from the diary server to the user system, the information comprising . . . configuration information for controlling behavior of a cohesive diary page, the configuration information including privacy level information.”).) The patent explains that **“configuration information”** and **“privacy level information”** refer to information used to determine whether certain content will be displayed to a user viewing or accessing a cohesive diary page. “If a user does not have sufficient permission to view an object in a diary,” for example, “the diary may not make the object visible to the user, i.e., the user does not even know that the object exists, or it may present the object using an

alternate representation.” (’316, 5:63-67.) This is accomplished through use of “configuration information,” such as passwords, privacy levels or information identifying the authorized user. (*Id.*, 9:2-4 (“The configuration information, such as privacy level, passwords, or the full name of the user, is user-specific.”), 10:51-54 (“The passwords for a particular user’s diary pages are stored as configuration information in that user’s diary information **114.**”); 17:22-24 (“User settings are stored as configuration data, which includes the passwords required for users to access the different privacy levels.”).) The related term “**privacy level information**” refers to configuration information that describes or specifies at least one user permitted to view particular content on a cohesive diary page. (’316, 2:40-44 (“The diary owner can set various levels of privacy for different portions of his diary. Thus, certain portions of the diary (for example, a daily entry or a reminder list) can by [sic] viewed only be [sic] the diary owner, while other portions of the diary can be viewed by anyone with a Web browser.”).)

The district court in the pending litigation identified in **Part I.B** above issued a claim construction ruling clarifying that “privacy level information” need not specify the “metes and bounds” or “universe” of users that have permission to access the content. (Ex. 1003, at 007 (“According to the Court’s construction, configuration information that describes a user who is permitted to view

particular content is privacy level information.”.) Accordingly, the Petitioner proposes that these terms be construed as:

- “configuration information for controlling behavior of a cohesive diary page” – information that determines what information will be displayed to a user who is viewing a cohesive diary page; and
- “privacy level information” – configuration information that describes or specifies at least one user permitted to view particular content on a cohesive diary page.

**VII. CLAIMS 1, 4, 17, 18, 20 AND 26 OF THE ’316 PATENT ARE UNPATENTABLE**

**A. Ground 1 – Claims 1, 4, 17, 18 and 26 are Obvious Over Salas In View of Tittel (1997) and Parker**

The alleged inventions of claims 1, 4, 17, 18 and 26 are obvious over U.S. Patent No. 6,233,600 B1 to Pito Salas et al. (“Salas”) [Ex. 1005] in view of Ed Tittel et al., *More HTML for Dummies* (2d ed. 1997) (“Tittel (1997)”) [Ex. 1006],<sup>2</sup> and U.S. Patent No. 5,729,734 to Robert D. Parker et al. (“Parker”) [Ex. 1011]. Salas and Parker qualify as prior art under 35 U.S.C. § 102(e) (pre-AIA) because they

---

<sup>2</sup> The Tittel reference cited in the text is referred to as “Tittel (1997)” to avoid confusion, as this Petition also includes as an exhibit (Ex. 1007) excerpts of an earlier 1995 book by Mr. Tittel that is used for background purposes.

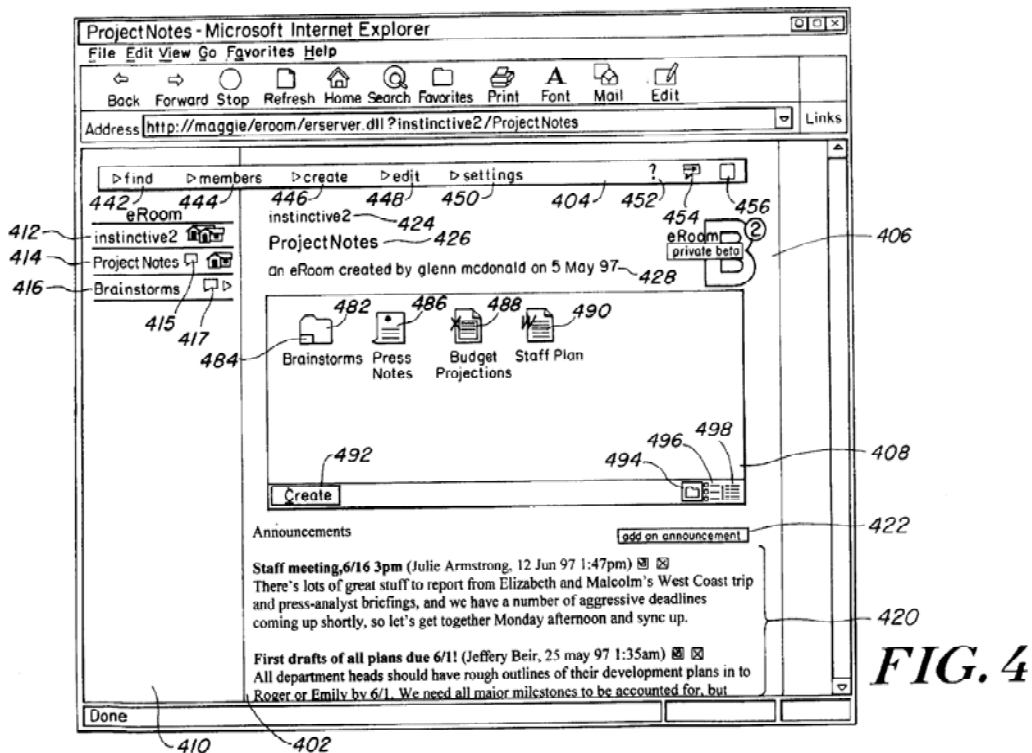
issued from applications filed prior to the filing date of the '316 patent (September 1, 1998). Tittel (1997) qualifies as prior art under at least § 102(a) and § 102(b) (pre-AIA) because it was published prior to the '316 filing date.

This Petition relies on Salas for the vast majority of the limitations in claims 1, 4, 17, 18 and 26. Salas is a sufficiently strong reference, in fact, that it could have been mapped as an anticipatory reference against these claims. But the Petitioner is mindful of the Board's desire to avoid potentially redundant grounds, including multiple grounds based on one or more of the same prior art references. The Petitioner has accordingly proposed an all-inclusive ground based on obviousness for these claims to reduce the burden on the Board.

**1. Claim 1 is Obvious Over Salas In View of Tittel (1997) and Parker**

Although the preamble of claim 1, "[a] method of organizing information for display," does not appear to recite a claim limitation, Salas nonetheless discloses it. Salas, entitled "Method and System for Providing a Networked Collaborative Work Environment," describes a technology known as an "eRoom" that provides a web-based system for organizing information. As explained in Salas, "[a]n eRoom is a set of connected HTML pages displayed to a user that displays project-related files, data and discussion lists." (Salas, 5:6-8.) Figure 4

shows an exemplary eRoom page displayed a web browser (*id.*, 5:21-24):



(*Id.*, Fig. 4.) Additional details about the eRoom system are provided in the claim limitations that follow.

**a. Claim 1[a] – “Diary Program”**

The first limitation of claim 1 recites the step of “sending from a diary server to a user system, a diary program capable of being executed by a browser in the user system.” Salas discloses a “diary server” in the form of an eRoom server (14) that “stores information relating to a project or a set of projects, referred to as a facility, in a database 20.” (Salas, 3:24-27; Fig. 1 (14).)

Salas also discloses a “user system” in the form of at least one client

workstation (**12**), connected to the server (**14**) through a network. “Client workstations **12**’ are connected to one or more servers **14**.” (*Id.*, 3:17-18.) “The network connecting the client workstations **12** and the server **14** may use any physical media, including wireless, provided that the physical media supports the HyperText Transfer Protocol (HTTP).” (*Id.*, 3:20-23.) Salas further explains that the eRoom server (**14**) sends information to the client workstation (**12**), where it can be stored. “Client workstations **12**’ generally have local memory elements for storing data objects of files associated with a project that are downloaded from the server **14** as well as one or more eRoom templates.” (*Id.*, 5:12-15.)

Salas discloses a “diary program” in the form of a software program executed by the user’s web browser that generates the HTML for the eRoom page (the “**cohesive diary page**”). (Tittel Decl. ¶ 56.) Salas describes, for example, “a method for locally generating HTML pages for display,” based on project data and a HTML web page template. (*Id.*, 2:20-27, 6:64-7:4.) The software that generates the HTML for the eRoom pages is referred to as a “page builder” application:

Generation, display and management of eRooms is controlled by a “page builder” application residing on the client workstation **12**’. The page builder application may be provided as specialized hardware or software operating on a general purpose machine. In some embodiments, the page builder application may be

provided as an Active X control or a COM object.

(*Id.*, 6:57-63; *see also* Tittel Decl. ¶ 56.)

Salas discloses other software programs executing in the client workstation (e.g. ActiveX controls) that generate and handle other aspects of the eRoom page. (Salas, 5:29-36 (describing ActiveX control for handling embedded discussions), 5:54-59 (describing ActiveX control for providing pictorial identification of the page); 13:23-26 (describing ActiveX control for handling uploading of files).)

The page builder and other ActiveX software programs described in Salas collectively correspond to the “**diary program**” recited in the ’316 patent. This is consistent with the ’316 patent, which specifically states that a diary program may be implemented using ActiveX controls. (’316, 6:53-56 (“[A]ny appropriate executable program can be used to implement the functionality of the diary applet **112**, including but not limited to JavaScript, ActiveX controls, Visual Basic, and plug-ins.”) (underlining added).)

The eRoom page generated by the ActiveX control (e.g. as shown in Fig. 4) qualifies as a “**cohesive diary page**.” Users interact with eRooms “by using Web browsers in a traditional manner. That is, users may traverse a hyperlink to access an eRoom, or users may directly enter a URL address into the browser.” (*Id.*, 6:40-43.) As explained in **Part VI.D**, above, the “cohesive diary page” of the ’316 patent



is not limited to a personal and private journal viewed by a single individual. The diary of the '316 patent, like an eRoom page, "can be read concurrently by multiple users in different locations." ('316, 5:48-49.) A diary may also contain placed on the diary page by its owner, or by any other user who has permission to do so – just like an eRoom page. (*Id.*, 5:19, 6:9-11.) The eRoom page in Salas therefore meets the requirements of a "cohesive diary page." (See Tittel Decl. ¶ 54.) The assembly of the cohesive diary page is further discussed in the discussion of claim 1[c] below.

Salas does not state that the eRoom server (14) sends the ActiveX controls to the client workstation (12). Salas instead simply takes for granted that the ActiveX controls are installed on the client workstation, and does not explain how they were delivered. As explained in the Tittel Declaration, this simply represents the omission of a trivial, well-understood detail that would have been readily apparent to one of ordinary skill in the art. (Tittel Decl. ¶ 58.) It was well-known that ActiveX controls were downloaded to the user's computer from the web server. (*Id.* ¶¶ 28-29, 58.) In order to embed an ActiveX control in a web page, the developer was required to expressly insert the URL from which the ActiveX file could be downloaded into the web page's HTML. (*Id.* ¶¶ 28-29 (description of <OBJECT> tag), ¶ 58.) One of ordinary skill in the art would therefore have

understood that the ActiveX controls in Salas were necessarily downloaded (at least once) from the eRoom web server. (*Id.* ¶ 59.) The ability to download an ActiveX control from the server hosting the web page was so basic to the functionality of ActiveX that one of ordinary skill in the art would have regarded Salas's absence of an express statement to that effect as insignificant, and in fact, as an omission of a detail within the general knowledge of a skilled artisan. (*Id.*)

Even assuming there was some question as to whether Salas inherently discloses that the ActiveX controls are sent by the eRoom server (**14**) to the client workstation (**12**), it would have been obvious to adapt Salas to require the client workstations to obtain the ActiveX controls from the server. As explained in Tittel (1997): "ActiveX objects may reside in stand-alone machines, on a LAN or intranet, or on the Internet. Specialized for the Internet, ActiveX objects are automatically downloaded if the object is not in the user's machine." (Tittel (1997), at 341 (definition of "**ActiveX**") (underlining added); *id.* at 176.) The ability to download an ActiveX control from the web server was built into the <OBJECT> element itself – the very technique by which an ActiveX control was embedded in a page. (Tittel Decl. ¶¶ 28-29, 60.) Adapting these teachings would have predictably resulted in the system of Salas in which the ActiveX controls were downloaded to the client workstation (**12**).

A skilled artisan as of September 1998 would have recognized a number of benefits of downloading ActiveX controls from the web server. First and foremost, it makes installation of ActiveX controls both easy and automatic for users. Software installation prior to the Internet and Web typically involved a user obtaining a CD-ROM or other physical medium and inserting it into the disc drive of the computer, in some cases requiring the assistance of a system administrator. (*Id.* ¶ 61.) These hassles and inconveniences are eliminated by the ability of an ActiveX control to download and install automatically upon the first visit to the web page on which the control has been placed. (*Id.*)

Additionally, using ActiveX download techniques ensured that the user always had the most up-to-date version of the ActiveX control or component. (*Id.* ¶ 62.) ActiveX downloading also provided a performance advantage in that the ActiveX control or component would only be downloaded once to the user's web browser (*e.g.* the first time the user logged onto the page). (*Id.*)

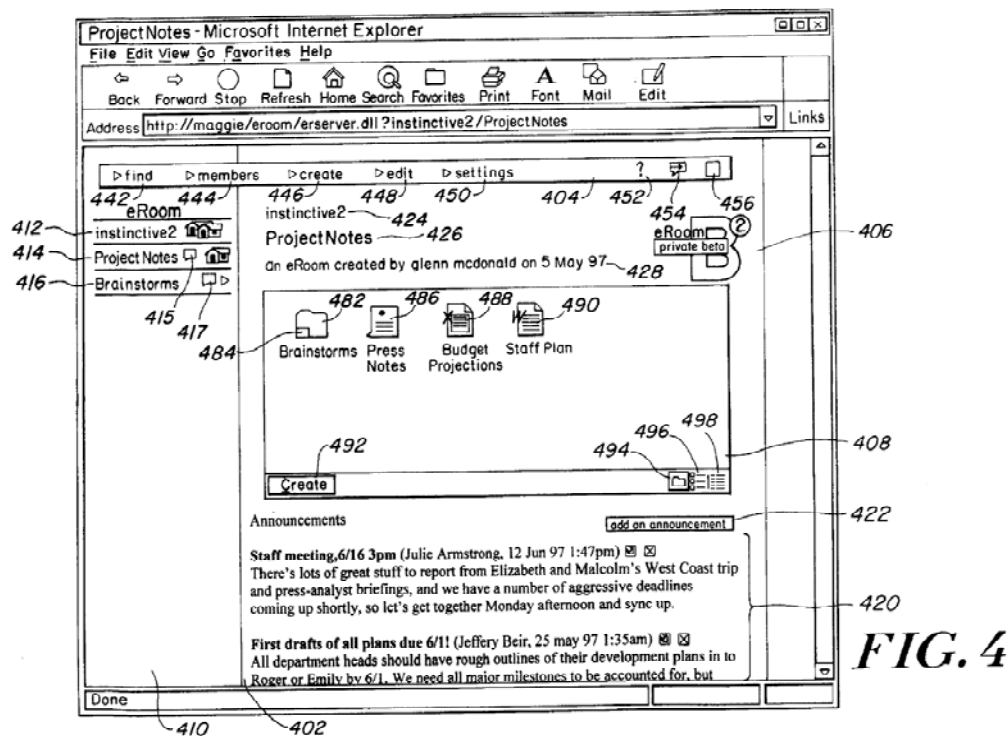
**b. Claim 1[b] – “Diary Information”**

The next limitation of claim 1 recites the step of “sending diary information from the diary server to the user system, the information comprising content data including an associated time, a page design to specify the presentation of the content data, and configuration information for controlling behavior of a cohesive

diary page, the configuration information including privacy level information.” This limitation requires the transmission of “diary information” that comprises three different types of information: **(1)** “content data including an associated time,” **(2)** “a page design,” and **(3)** “configuration information . . . including privacy level information.” The Petition will address each component below.

**(1) “content data including an associated time”**

Each eRoom project contains a “project database” that is stored on the eRoom server **(14)**. (Salas, 2:30-32, 3:24-27, Fig. 1 **(14)**.) The client workstation **(12)** stores some subset of the project database, such as “information related to the project which is relevant to that user’s effort on the project.” (*Id.*, 2:32-34.) Those project files were downloaded from the server. As explained in Salas: “Client workstations **12’** generally have local memory elements for storing data objects of files associated with a project that are downloaded from the server **14** as well as one or more eRoom templates.” (Salas, 5:12-15 (underlining).) “The client workstation **12’** uses project data received from the server **14** in combination with one or more template files to create and display to the user . . . a private, secure collection of HTML pages . . . .” (*Id.*, 4:66-5:2.) These content objects are represented in Figure 4 below depicting an eRoom:



(Salas, Fig. 4.) As shown in Figure 4 above, the content data includes items such as entries under “Announcements,” the first being a “Staff meeting” entry from “Julie Armstrong” at “12 Jun 97 1:47 pm.” (*Id.* (underlining added).)

Salas discloses at least two other examples of transmission of content data that includes an associated time. Figure 4 also includes item box (408) in the middle of the page that contains icons for content items, including a notes file “Press Notes” (486), a spreadsheet file “Budget Projections” (488), and a word processing file “Staff Plan” (490). (*Id.*, 5:60-64.) Each item has an associated time. Although those dates or times are not visible in Figure 4, this is because the current display mode for the item box (408) in Figure 4 happens to be set to “icon

display,” which displays only the icon and name of the item and does not display the date and time. (*Id.*, 6:4-7.) The eRoom page can display files using three different user-selectable viewing modes, one of which provides a list display that shows the date and time associated with each file. (Tittel Decl. ¶ 67.)

In particular, Salas explains that: “In the embodiment shown in FIG. 4, three icons are provided: an ‘icon display’ icon **494** (currently selected) which causes items to be displayed as large icons with identifying text underneath; a ‘list display’ icon **496** which causes items to be displayed as small icons with identifying text to one side of the icon; and a ‘report display’ icon which causes items to be displayed as a list.” (*Id.*, 6:4-10 (emphasis added).) Selecting the “report display” icon will produce a list of the files in the item box (**408**), which “may be alphabetized, ordered by size of item, ordered by creation date, ordered by modification data [sic; date], or ordered by some other data field associated with each item.” (*Id.*, 6:10-13.) Salas therefore makes clear that each of the files in the item box (**408**) comprises content data that includes an associated time of creation and modification, even though that information was not shown in the “icon mode” display used to create Figure 4. The ability to display the creation and modification dates in the “report display” mode indicates that date information was transmitted to the client workstation (**12**). (Tittel Decl. ¶ 68.)

Finally, Salas discloses a technique for displaying the modification time associated with each content item without the user having to change display modes in the item box (408). In particular, an eRoom can be configured to automatically display the content items in “report display” mode and to include columns showing the modification time associated with each item. (Salas, 8:34-38 (“The ‘DYNAMIC’ section for the example above specifies that the item box in a Folder starts displaying items in Report display, does not show a column for creation date, does show the time along with the date in the Modified column, and has modified column widths.”) (underlining added).) This provides yet another example in which content data and an associated time is sent from the server (14) to the client workstation (12). Any one of these examples would be sufficient to teach sending, from the server to the user system, content data including an associated time. (Tittel Decl. ¶ 69.)

**(2) “a page design”**

Salas also discloses that the eRoom server (14) sends “diary information” that includes “a page design to specify the presentation of the content data.” The “page design” information in Salas comprises a “wrapper” file which, in turn, identifies a “template” that specifies the layout and style of the eRoom page. The “wrapper” file is retrieved from the server when the user accesses an eRoom by

entering the URL of the eRoom into a web browser. “Regardless of the manner in which the URL address is accessed, the browser retrieves the HTML file in order to display it. However, if the URL address is an eRoom, the server of the file returns a file of control information, referred to as a ‘wrapper’ file.” (Salas, 6:43-47 (underlining added).) This “wrapper” file identifies the eRoom template the page builder (ActiveX control) will use to generate the eRoom page:

Referring to FIG. 5, the first step that is taken by the page builder application is to retrieve the eRoom template indicated by the wrapper file (step 502). An eRoom template is an HTML file having additional eRoom-specific information embedded in it.

(*Id.*, 6:64-7:1 (underlining added).)

The eRoom template contains layout or style information that defines, at least in part, the visual appearance of the eRoom page. According to Salas, “eRoom information in a template includes sections controlling the page itself, the controls on it, and the way the page’s data is presented [sic; when] the page is created or edited.” (*Id.*, 7:8-10.) The eRoom template includes a series of “properties” specifying, for example, the way the items will be displayed on the page (e.g., in small icon, large icon or report view modes) (*id.*, 8:55-58), the width of the item box in Figure 4 (*id.*, 9:31-35), whether particular pieces of information



will be displayed (e.g., the size, owner, date of a file) (*id.*, 9:13-24), and how columns on the eRoom page should be sorted (*id.*, 9:13-30). The “wrapper” and “template” files are independent of the content data, as they are separate files and nothing in Salas suggesting that they are embedded within the content files themselves. (Tittel Decl. ¶ 71.)

Salas makes clear that the “wrapper” file is transmitted from the server (24) to the client workstation (12) (Salas, 6:45-47), but reminiscent of the “**diary program**” discussion above, Salas does not specify how the eRoom template made its way onto the client workstation. Salas states that the “wrapper” file was downloaded from the server (*id.*, 6:45-47), but the “wrapper” file does not itself appear to carry the eRoom template. (Tittel Decl. ¶ 72.) But transmission of the “wrapper” file is itself sufficient to satisfy this claim limitation under its broadest reasonable construction. This is because the “wrapper” file identifies the corresponding page design.

Even if the claim language required physical transmission of the actual page design data itself, adapting the eRoom server (14) of Salas to transmit both the wrapper file and the eRoom template would have been obvious to one of ordinary skill in the art. (Tittel Decl. ¶ 73.) The ability to reference another file in the HTML of a web page, which would cause the web browser to download that

file from the server hosting the web page, was well-known to persons of ordinary skill in the art. With respect to page design information (*e.g.* style sheets), in fact, a technique for doing this had already been standardized by 1997. (Tittel Decl. ¶¶ 20-23.) Style sheets serve substantially the same purpose as the templates in Salas – to specify a set of parameters for the appearance of a page. (See Tittel (1997), at 58 (“In general, a style sheet defines a set of layout parameters for a document to ensure that similar elements in the document appear uniformly.”).)

Accordingly, it would have been obvious to one of ordinary skill in the art to adapt the “wrapper” file of Salas to include a URL identifying an eRoom template stored on the eRoom server (**14**). Salas makes clear that the template file itself is an HTML file. (Salas, 6:66-7:1 (“An eRoom template is an HTML file having additional eRoom-specific information embedded in it.”) (underlining added).) The wrapper thus certainly could have included the <LINK> tag discussed in **Part IV.A** above, which was a standard HTML tag at the time of the alleged invention. (Tittel Decl. ¶ 74.) Adding a <LINK> tag to the wrapper HTML file in Salas to identify the location of a template file on the eRoom server (**14**) would have been trivial to implement, and would have predictably resulted in the system of Salas in which the wrapper file *and* the eRoom template file were downloaded from the server to the client when the eRoom is accessed by the web browser. (*Id.*)

Downloading the template file from the eRoom server (14) would also have presented several advantages over a system in which the template files must be stored or “pre-loaded” onto the client workstation. (*Id.* ¶ 75.) Many of these advantages are the same as those discussed above for downloading the ActiveX components (“**diary program**”) from the server. Salas indicates that the template file is used by the ActiveX control to generate the eRoom web page. (Salas, 6:57-7:1, 10:66-11:3.) The benefits of obtaining the eRoom template from the server, therefore, mirror the advantages of obtaining the ActiveX controls from the server, including: (1) simplifying installation of the ActiveX controls and templates; (2) ensuring that the user has the most up-to-date and correct template for use with the ActiveX controls; and (3) the ability to only download the template once to the user’s web browser. (Tittel ¶ 75.) Downloading the template files from the server (14) would also allow the user to access an eRoom from a different computer on which the eRoom template files have not yet been installed. This would provide flexibility and reduce administrative costs by avoiding the need to “pre-install” the template files on each computer. (*Id.*)

### (3) “configuration information”

Finally, Salas discloses that the eRoom server (14) sends “diary information” that includes “configuration information for controlling behavior of

a cohesive diary page, the configuration information including privacy level information.” As explained above in **Part VI.E** above, “configuration information” and “privacy level information” are used to determine whether a user is permitted to view or access content on a diary page.

Salas explains that “[s]ince every file and eRoom is represented as an object in the database, access of users to each item can be controlled by entries in the database schema.” (Salas, 14:37-39.) For example, each file in the database may contain “file metadata” that includes “access information such as which users may open, view and edit the file.” (*Id.*, 13:52-55 (underlining added).) This access information can specify whether the requesting user has permission to view and/or edit a particular file. (*Id.*, 12:34-39 (“[T]he access rights of the requesting user are checked. If the user has appropriate access rights, i.e., ‘can edit’ if the user has indicated editing will occur or ‘can view’ if the user has indicated only viewing will occur, the user will be allowed to retrieve the file.”).)

Salas also makes clear that this privacy level information is transmitted to the client workstation (**12**) prior to the display of the eRoom page. Salas describes a file synchronization technique in which the server sends updated information to the client workstation. (*Id.*, 11:54-60.) This synchronization results in the file metadata – which includes the access rights discussed above – being transmitted

to the user system. “Once synchronization has been accomplished and local database metadata has been updated, the appropriate data objects are inserted into the eRoom . . . .” (*Id.*, 12:1-3.) The fact that the file metadata is sent to the user system prior to display of the eRoom page is also shown in the fact that the file metadata includes the name of the file (*id.*, 13:52), which is displayed on the eRoom page. (*Id.*, Fig. 4.) As explained above, the ability to display the date and/or time of files on the eRoom page also indicates that the file metadata for was sent by the server to the client workstation. (Tittel Decl. ¶¶ 68, 78.)

**c. Claim 1[c] – Assembling the “Diary Page”**

Claim 1, limitation [c], recites the step of “assembling the cohesive diary page by dynamically combining the content data and the page design in accordance with the configuration information for the cohesive diary page to be displayed by the diary program running in the browser.” As mentioned previously, the “**cohesive diary page**” in Salas is the eRoom page, an example of which is shown in Figure 4. (Salas, Fig. 4.) The page builder application in Salas (“**diary program**”) assembles the eRoom page (the “**cohesive diary page**”) by combining the project data (“**content data**”) and the eRoom template (“**page design**”):

The client workstation 12’ uses project data received from the server 14 in combination with one or more template files to create and

display to the user of the client workstation a private, secure collection of HTML pages that provide a virtual workroom for members of a team, whatever its size and wherever the members of the team are physically or corporately located, may be referred to throughout as an “eRoom”, or an “eRoom page”.

(*Id.*, 4:66-5:6; *see also id.*, 6:57-59 (“Generation, display and management of eRooms is controlled by a ‘page builder’ application residing on the client workstation **12’.**”) (underlining added).) Salas explains that the page builder application generates the web page by, among other things, “filling in” replacement properties specified in the eRoom template. (*Id.*, 10:66-11:3.) The process for performing this replacement, in fact, closely mirrors the technique disclosed in the ’316 patent. (Tittel Decl. ¶¶ 79-80.)

As explained previously, Salas discloses privacy level information in the form of “file metadata” that includes “access information such as which users may open, view and edit the file.” (Salas, 13:52-55.) The displayed eRoom page ensures that each user can access the files based on his or her specific access control setting. As explained in Salas: “Files displayed by an eRoom may be viewed or edited by team members having the appropriate access controls (discussed in more detail below).” (*Id.*, 12:7-9.) The eRoom page in Salas is therefore assembled “in accordance with the configuration information for the

cohesive diary page,” as recited in the ’316 patent.

As explained later in that column of Salas, when a user viewing an eRoom page attempts to access a file, the user’s access level is verified to ensure that the user can perform the requested operation:

[T]he access rights of the requesting user are checked. If the user has appropriate access rights, i.e., “can edit” if the user has indicated editing will occur or “can view” if the user has indicated only viewing will occur, the user will be allowed to retrieve the file.

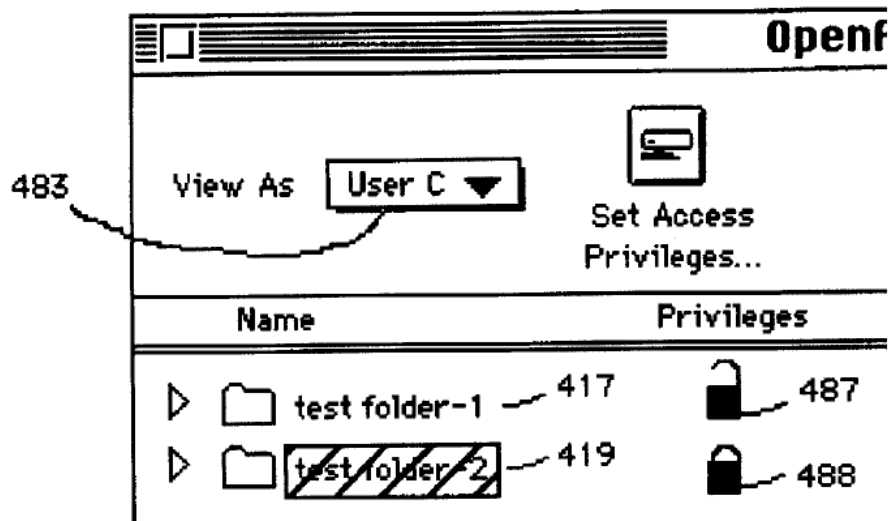
(*Id.*, 12:34-39.) Salas makes clear, therefore, that the system allows the user to view the list of files in the project but the user may be certain types of access to the file. The access rights verification step described above is performed by a software process (“daemon”) on the client workstation. (*Id.*, 12:16-22, 12:23-27.)

The disclosures from Salas discussed above, including the statement that “[f]iles displayed by an eRoom may be viewed or edited by team members having the appropriate access controls” (*id.*, 12:7-9), by itself suffice to disclose the assembly of the cohesive diary page “in accordance with the configuration information,” as recited in claim 1. (Tittel Decl. ¶ 83.)

If claim 1 is interpreted to require some kind of visual indication on the “cohesive diary page” of the privacy level associated with the viewing user (*id.* ¶¶

83, 84), the disclosures of Salas could be combined with U.S. Patent No. 5,729,734 to Robert D. Parker et al. (“Parker”) [Ex. 1011] to render this limitation obvious to one of ordinary skill in the art. Parker, entitled “File Privilege Administration Apparatus and Methods,” discloses a series of user interface techniques to manage user access privileges for a server “sharepoint,” which is defined in Parker as “an item, e.g., a file, a folder, a volume, a hard disk, or the like, that is capable of being shared by the network users.” (Parker, 2:24-26.)

Parker explains that each file or item may be displayed to a user in a way that visually indicates the privilege or access level granted to that user. This is shown in Figure 7, which shows a folder listing as it would appear to “User C”:



(*Id.*, Fig. 7 (partial figure shown).)

Parker explains that the display in Figure 7 shows precisely how “User C”



would see the two folders (417, 419) on its screen. (*Id.*, 11:32-35 (“When the administrator views as user C, the administrator is viewing from user C’s perspective (i.e., in accordance with user C’s access privileges as if the administrator was sitting at user C’s client terminal).”).) In the example above, because the administrator did not grant “User C” rights to “test folder-2” (419), that folder appears grayed out on the screen with a "lock" icon (488):

Since the administrator granted user C read access privileges to test folder-1 (417) in FIG. 5, test folder-1 (417) is shown to be accessible in FIG. 7 (by open lock icon 487 and by the fact that “test folder-1” is not ghosted, grayed out, or hidden). On the other hand, test folder-2 (419) is in a ghosted state indicating no access to test folder-2. In other embodiments, test folder-2 (419) may be grayed out, shaded, or hidden altogether.

(*Id.*, 11:40-47 (underlining added).) This is consistent with the ’316 patent, which states that a diary page “may not make the object visible to the user, i.e., the user does not even know that the object exists, or it may present the object using an alternate representation.” (’316, 5:64-67.)

Parker therefore discloses the creation of a page displayed “in accordance with the configuration information” including “privacy level information,” as recited in claim 1[c]. Parker specifically states Figure 7 shows a display “from user

C's perspective (i.e., in accordance with user C's access privileges . . .).” (Parker, 11:32-35 (underlining added for emphasis).) Limitation [c] of claim 1 is therefore disclosed by Salas in view of Parker. (Tittel Decl. ¶ 87.)

It would have been obvious to add the user interface features of Parker to Salas. Both references disclose similar systems for organizing and displaying files in a networked environment that have access restrictions. (Salas, 12:7-9, 13:52-57; Parker, 1:19-28.) Both systems also employ a similar user interface for displaying files or folders in icon or list form on the user's screen. (Salas, Fig. 4 (**482, 486, 488, 490**); Parker, Fig. 7 (**487, 488**).) It would have been obvious to one of ordinary skill in the art to add, to the eRoom system of Salas, the user interface technique of Parker in which an item is displayed normally (if access is allowed) or “in a ghosted state,” “greyed out, shaded, or hidden altogether” (Parker, 11:44-47), depending on the viewing user's access privileges. This would predictably result in the eRoom system of Salas in which the ActiveX control running in the web browser obtains the file metadata identifying the access privileges of the user (Salas, 13:52-57), and then uses it to generate the eRoom page with an item box (**408**) listing each file with a visual indication consistent with the viewing user's access privileges (*e.g.* fully visible, ghosted, grayed out, with a “lock icon,” etc.). (Tittel Decl. ¶ 88.)

A person of ordinary skill in the art would have had several motivations to make this combination. Parker itself provides a motivation by stating that “it is envisioned that a user accessing the network via its user privilege, will be able to distinguish whether the listed sharepoints are inaccessible, or accessible for read & write, read-only or write only via various access indicating icons.” (Parker, 11:50-54 (underlining added).) The addition of this known element from Parker would provide an improvement to Salas by allowing a user to visually ascertain, before attempting to access a particular item listed in an eRoom, whether the requested access is permitted. (Tittel Decl. ¶ 89.) This would also result in a more streamlined user experience in that the user could preemptively avoid attempting file accesses that were not permitted. (*Id.*)

**d. Claim 1[d] – Change of “Diary Information”**

The fourth limitation of claim 1 recites the step of “receiving by the diary server at least one request for at least one change concerning the diary information, from the diary program in the user system.” Salas describes several features that independently disclose this limitation.

A user of an eRoom can change the project files and other content for the eRoom (“**diary information**”) by modifying an existing file or uploading a new file to the eRoom. For example, “users may perform work on files and objects locally

and upload the modified files and objects to the server **14** for viewing, comment, or further modification by other project team members.” (Salas, 13:7-10.) The user accomplishes this by “dragging” a file onto the eRoom:

Once the user is finished editing the file, it may be uploaded to the server **14** to allow other users access to it. The user signals that the file should be transmitted to the server **14** by dragging the file onto an eRoom displayed by the browser (step **704**). Dropping the file into the displayed eRoom invokes an ActiveX control or a background daemon process which manages the upload of the file to the server **14**.

(*Id.*, 13:19-26 (underlining added).)

Salas provides another example in which a user can contribute to an on-line discussion, such as the “Announcements” (**420**) shown in Figure 4. The announcements feature permits the user to add to the discussion:

In the embodiment depicted in FIG. 4, a discussion **420** is embedded within the page and there is a facility **422** to allow a viewer to contribute to the discussion **420**. The embedded discussion **420** and the contribution facility **422** may be implemented as ActiveX controls, a JAVA applet, or various other means.

(*Id.*, 5:28-34 (underlining added).)

As explained for claim 1[a], the ActiveX controls identified in the quotations above comprise the recited “diary program.” The “request for at least one change

concerning the diary information” in each embodiment, therefore, comes “from the diary program” as recited in claim 1[d].

**e. Claim 1[e] – Sending “New Diary Information”**

The final limitation of claim 1, limitation [e], recites the step of “sending, by the diary server to the user system, new diary information for changing the cohesive diary page.” Salas explains that, from time to time, the contents of the eRoom and the eRoom page need to be updated. “Put another way, the client workstation’s local project database 20’ must be synchronized with the server’s project database 20 to ensure data coherency.” (Salas, 11:16-19.)

Salas discloses several techniques for performing synchronization. (*Id.*, 11:19-27.) Each of these techniques relies on a “modification tag value” that is used by to determine whether an item on the local database is up-to-date. (*Id.*, 11:27-30, 11:42-47.) “If the modification tag value or state bits indicate that the object needs to be synchronized,” Salas explains, “the updated object may be requested from the server in the foreground or in the background.” (*Id.*, 11:30-34; *see also id.*, 11:34-41.)

Once this “new diary information” is received by the client workstation, it may be used to change the cohesive diary page. “Once synchronization has been accomplished and local database metadata has been updated, the appropriate

data objects and values are inserted into the eRoom where indicated by Replace Properties, and the eRoom is displayed to the user (step 508) by the browser application in a traditional manner (refer to FIG. 4).” (*Id.*, 12:1-6 (underlining added).) Salas therefore discloses “sending, by the diary server to the user system, new diary information for changing the cohesive diary page.”

**2. Claim 4 is Obvious Over Salas In View of Tittel (1997) and Parker**

Dependent claim 4 recites: “The method of claim 1 wherein the new diary information is for changing content of the diary page without changing a general appearance of the diary page.” The term “**new diary information**” appears to refer back to limitation [e] of claim 1 reciting the step of “sending, by the diary server to the user system, new diary information for changing the cohesive diary page.” Salas discloses this additional requirement.

As I explained above for claim 1[d] and 1[e], Salas discloses that a user can change the “diary information” by, among other ways, adding to the “Discussion” portion of an eRoom or uploading a new file or modifying an existing file. One of ordinary skill in the art would understand that the result of these actions would be to add to or modify entries on the eRoom page, but leave the overall look and appearance of the eRoom page the same. (Tittel Decl. ¶ 98.) This is because the

appearance of the eRoom page is determined by the “template” that specifies how the page will appear to the user. (Salas, e.g., 7:8-10.)

For example, if a user uploaded a new file to the eRoom pictured in Figure 4, for example, this would result in the new file being inserted into the item box (408), perhaps as an additional icon, after the client workstation’s local database was synchronized. (*Id.*, 12:1-6.) Similarly, if a user contributed new content to the Announcements section (422) (*id.*, 5:29-37), this would result in the comment being added to that section. In either case, placement of the content on the page is determined by “filling in” the content in the locations specified by the template. (*Id.*, 10:66-11:3, 10:44-49.) With respect to the addition of new data to the eRoom, for example, “[o]nce synchronization has been accomplished . . . the appropriate data objects and values are inserted into the eRoom where indicated by Replace Properties, and the eRoom is displayed to the user (step 508) by the browser application in a traditional manner (refer to FIG. 4).” (*Id.*, 12:1-6 (underlining added).) The “Replace Properties” are determined by the template, which does not change with a change in content. Salas therefore discloses that the new diary information is for changing content of the diary page without changing its general appearance, as stated in claim 4.

**3. Claims 17 and 26 Are Obvious Over Salas In View of Tittel (1997) and Parker for the Same Reasons as Claim 1**

As explained in **Part V.B** above, independent claim 17 is similar in many respects to claim 1 except that it is written as an apparatus claim that recites “a software portion configured to” perform substantially the same steps as claim limitations **1[a]**, **1[b]** and **1[c]**. Claim 26 depends from claim 17 and recites functions substantially similar to claim limitations **1[d]** and **1[e]** relating to changing diary information and receiving new diary information.

The recitation of a “software portion” in the limitations of claims 17 and 26 does not provide a basis to distinguish the disclosures of Salas discussed above. One of ordinary skill in the art would understand that the eRoom system in Salas, in order to perform the functions recited in claims 17 and 26, would necessarily include software executing on the client workstation (**12**) that was programmed to carry out those functions. (Tittel Decl. ¶ 102.)

Salas also expressly discloses that software. Claim 17**[a]** recites “a software portion configured to receive, by a user system from a diary server, a diary program capable of being run by a browser in the user system.” This function is performed in Salas for all of the reasons provided for claim **1[a]** above. The “software portion” here takes the form of a web browser installed on the user’s



client workstation. (Salas, 5:21-24.) As explained in Salas, “[t]he browser executing on the client workstation interprets the HTML file that it has received and displays the Web page to the user of the client workstation.” (*Id.*, 1:31-34; *see also id.*, 6:40-43.) Because the ActiveX controls that make up the “diary program” in Salas may be embedded in an HTML file (see **Part III.B**, above), which is received and interpreted by the web browser, the browser performs this function of receiving the diary program as recited in claim 17**[a]**. This is consistent with the way this step is described in the ’316 specification. (’316, 8:35-38 (user starts web browser), 8:42-46 (web browser receives and executes diary program).)

Claim 17**[b]** recites in relevant part “a software portion configured to receive, by the user system from the diary server, diary information . . .” Salas in view of Tittel (1997) discloses the receipt of the recited “diary information” for the reasons explained for claim 1**[b]** above. The “software portion” in Salas includes the web browser and ActiveX controls (including the “page builder application”). (Salas, 6:57-63, 6:64-7:7:2.) Salas also describes a process (“background daemon”) running on the client workstation (**12**) that handles the downloading of content from the eRoom server (**14**). (*Id.*, 11:11:12-26.) These software components perform at least the functions recited in claim 17**[b]**.

Claim 17**[c]** recites “a software portion configured to assemble the cohesive

diary page by dynamically combining the content data and the page design in accordance with the configuration information.” Salas in view of Parker discloses the assembly of the “cohesive diary page” for the reasons explained for claim 1[c] above. The “software portion” here would include the ActiveX control that implements the eRoom “page builder” application. (Salas, 6:57-63, 10:66-11:12.)

Claim 17[d] recites “a software portion configured to display the cohesive diary page, by the diary program running in the browser.” The “software portion” performing this function would comprise a combination of the “diary programs” (discussed in claim 1[a] above), and the web browser itself. (Salas, 12:4-6 (“[T]he eRoom is displayed to the user (step 508) by the browser application in a traditional manner (refer to FIG. 4).”) (underlining added), Fig. 4, 6:57-63.)

Dependent claim 26 recites “a portion configured to receive, from the diary server, new diary information,” followed by “a portion configured to change content of the diary page without changing a general appearance of the diary page, in accordance with the new diary information.” The software portions for performing these steps are substantially the same as those for claims 17[b], [c] and [d]. This includes at least the ActiveX page builder application (Salas, 6:57-63, 12:1-6), the ActiveX upload daemon (*id.*, 13:19-26), the ActiveX “Announcements” tool (*id.*, 5:28-34), and the web browser (*id.*, 12:1-6). Claims 17 and 26 are

therefore obvious over Salas In view of Tittel (1997) and Parker for the same reasons as claim 1 as discussed above. (Tittel Decl. ¶¶ 101-107.)

**4. Claim 18 Is Obvious Over Salas In View of Tittel (1997) and Parker**

Claim 18 depends from claim 17 and states: “The apparatus of claim **17**, further comprising: a portion configured to display contents of the diary page by the diary program in accordance with a cover for a diary.” It is not clear that this limitation adds anything different to the “page design” information described in claim 17. As explained in **Part V.A** above, the ’316 specification uses the term “cover” refer to the same information as the “page design.” (’316, 9:7-10 (“In step **308**, diary applet **112** generates one or more pages of the diary in HTML in accordance with the cover, content, and configuration information. The HTML is displayed as a diary page by browser **110**.”) (underlining added), 17:60-63 (“Covers are also called ‘presentation contexts.’ In one embodiment of the present invention, the presentation context can consist of HTML files in which ‘on the fly’ substitutions by the diary applet are performed.”).) The district court’s claim construction order gave identical meanings to “cover” and “page design.” (Order [**Ex. 1003**], at 002.) The software portion recited in claim 18, therefore, is disclosed in Salas for the same reasons as the software portion for the “assembly”

and “display” functions in claim 17[c] and 17[d] above. (Tittel Decl. ¶ 109.)

Even if the “cover” must include layout and style information different from the “page design” of claim 17, the eRoom template in Salas includes so many display, layout and style properties that one could simply select different pieces of information from that template to show the claimed “page design” and “cover.” (*Id.* ¶ 111.) One example of style information not identified in the discussion of “page design” above is “Wizard” information that describes the appearance of on-screen controls for creating and editing an eRoom page. (Salas, 9:38-43.) Another example of “cover” information is a “graphical element **406** [that] is used to pictorially identify the viewed page. The graphical element **406** may be a corporate logo or other organizational identifier.” (*Id.*, 5:54-59.) This graphical element may be a static image “or it may be a dynamic identifier such as a JAVA script or ActiveX control.” (*Id.*, 5:57-59.) In any event, Salas discloses a software portion “configured to display contents of the diary page by the diary program in accordance with a cover for a diary.” (Tittel Decl. ¶ 111.)

**B. Ground 2 – Claim 20 Is Obvious Salas In View of Tittel (1997) and Parker, and Further in View of Angles**

Claim 20 depends from claim 18 and recites: “The apparatus of claim **18**, wherein the cover includes advertisements not requested by a user.” This claim

Petition for *Inter Partes* Review of  
U.S. Patent No. 6,415,316

appears to add nothing more than the ability to add an advertisement to the diary page recited in claims 17 and 18. The ability to add advertisements to web pages was well-known prior to September 1998. (Tittel Decl. ¶ 113.)

One example of a Web advertising system is disclosed in U.S. Patent No. 5,933,811 to Paul Angles et al., entitled “System and Method for Delivering Customized Advertisements within Interactive Communication Systems” [Ex. **1012**]. Angles issued from an application filed on August 20, 1996, more than two years before the application for the ’316 patent. Angles discloses an on-line advertising service that can custom tailor specific advertisements to particular consumers. (Angles, 2:46-49.) Angles explains that “there has been a tremendous proliferation of corporate advertising across the Internet. For example, some companies such as Yahoo Corporation offer free services, such as the ability to search for particular sites on the Internet, but post advertising messages to consumers to help offset the cost of their service.” (*Id.*, 2:18-27.)

Angles discloses a system that selects an advertisement for the user and then incorporates it into a Web page. (*Id.*, 21:34-52.) The page “appears to the consumer like all other electronic pages **32** on the Internet **33**, except that it contains the customized advertisement **30** which has been pre-selected for that consumer.” (*Id.*, 21:49-52.) The selection of the advertisement is “based on the

consumer's profile" (*id.*, 8:58-61), not based on a request by the user.

It would have been obvious to one of ordinary skill in the art to adapt the system of Salas to display the eRoom page in accordance with a cover for a diary that included advertisements not requested by a user. The motivation for such a combination could not be any more clear and compelling – obtaining revenue. That revenue could be used for profit, to defray the costs associated with providing the web site, or to facilitate offering the web site at a reduced fee. (Tittel Decl. ¶ 117.) Angles expressly and repeatedly confirms these motivations. The system of Angles “allows the advertisement provider to monitor the number of advertisements viewed by consumers associated with a particular content provider. With this information, the content providers can receive advertising revenue based on the number of consumers who access their websites.” (*Id.*, 4:17-23; *see also id.*, 4:45-47 (“The Internet providers can then use this advertising revenue to reduce consumer access fees.”).)

The eRoom system of Salas could be readily adapted to include an advertisement as disclosed in Angles. The eRoom page includes a graphical element (406) displayed on the page. (Salas, Fig. 4 (406).) “The graphical element 406 may be a corporate logo or other organizational identifier. The graphical element may be static (as depicted in FIG. 4) or it may be a dynamic identifier

such as a JAVA script of ActiveX control.” (*Id.*, 5:55-59.) One of ordinary skill in the art would have recognized that an advertisement could be placed anywhere on the eRoom page, or could augment or even replace the “cover” information associated with the graphical element (406). (Tittel Decl. ¶ 119.) Adding web-based advertising from Angles to the eRoom system of Salas would predictably result in the system of Salas in which the eRoom page (Fig. 4) is displayed in accordance with a cover that includes an advertisement not requested by the user, as recited in claim 20. (*Id.*)

#### VIII. CONCLUSION

The Petitioner respectfully requests that the Board institute *inter partes* review of claims 1, 4, 17, 18, 20 and 26 of the ’316 patent, and find those claims unpatentable, based on the grounds presented in this Petition.

Dated: February 6, 2014

Respectfully submitted,

COOLEY LLP  
ATTN: Patent Group  
1299 Pennsylvania Ave., NW, Suite 700  
Washington, DC 20004  
Tel: (703) 456-8000  
Fax: (202) 842-7899

By: /Heidi L. Keefe/  
Heidi L. Keefe  
Reg. No. 40,673  
Counsel for Petitioner  
Facebook, Inc.

Petition for *Inter Partes* Review of  
U.S. Patent No. 6,415,316

**CERTIFICATE OF SERVICE**

I hereby certify, pursuant to 37 CFR sections 42.6 and 42.105, that a complete copy of the attached **PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 6,415,316, including all exhibits (Nos. 1001 through 1013)** and related documents, are being served via Federal Express on the 6th day of February, 2014, the same day as the filing of the above-identified document in the United States Patent and Trademark Office/Patent Trial and Appeal Board, upon the patent owner's attorneys of record:

Thomas Horstemeyer LLP  
400 Interstate North Parkway SE  
Suite 1500  
Atlanta, GA 30339

and upon the counsel of record for the patent owner in the litigation before the United States District Court for the Eastern District of Virginia, Case Docket No. 1:13-cv-158-TSE-TRJ entitled *Rembrandt Social Media LP v. Facebook, Inc., and AddThis, Inc.*:

Ahmed J. Davis  
Fish & Richardson PC  
1452 K Street N.W. 11<sup>th</sup> Floor  
Washington D.C. 20005

Robert E. Hillman  
Fish & Richardson PC  
One Marina Park Drive  
Boston MA 02210

Thomas Melsheimer  
Fish & Richardson PC  
1717 Main Street, Suite 5000  
Dallas, TX 75201

John S. Goetz  
Fish & Richardson PC  
601 Lexington Avenue 52d Floor  
New York, NY 10022

/ Heidi L. Keefe /  
Heidi L. Keefe  
Reg. No. 40,673



Petition for *Inter Partes* Review of  
U.S. Patent No. 6,415,316

COOLEY LLP  
ATTN: Heidi L. Keefe  
Patent Docketing  
1299 Pennsylvania Ave. NW  
Suite 700  
Washington, D.C. 20004  
Tel: (650) 843-5001  
Fax: (650) 849-7400