UNITED STATES PATENT AND TRADEMARK OFFICE

_____

BEFORE THE PATENT TRIAL AND APPEAL BOARD

_____

Facebook, Inc.
Petitioner

v.

Rembrandt Social Media, L.P.
Patent Owner

_____

IPR2014-00415
Patent 6,415,316

_____

DECLARATION OF MARK T. JONES, PH.D.

I, Mark T. Jones, declare:

1. I make this declaration at the request of the patent owner. Except as indicated herein, I have personal knowledge of the facts set forth in this declaration and, if called to testify as a witness, could and would competently testify to them under oath.

2. I am being compensated at the rate of $450 per hour plus expenses for time spent in connection with this declaration. My compensation does not depend upon the opinions I render or the outcome of this litigation.

3. I am a Professor of Electrical and Computer Engineering at Virginia Tech in Blacksburg Virginia. I graduated summa cum laude from Clemson University in 1986 with a B.S. in Computer Science and a minor in Computer Engineering while holding a National Merit Scholarship and the R. F. Poole Scholarship. I then graduated from Duke University in 1990 with a PhD in Computer Science while holding the Von Neumann Fellowship.

4. Upon graduation, I joined the Department of Energy at their Argonne National Laboratory facility. My responsibilities there included the design and use of software for computers with hundreds of processing elements. This software was designed for compatibility with new parallel computer architectures as they became available as well as with other large software components being written in the Department of Energy. While with DOE, I received the IEEE Gordon Bell Prize.

5. In 1994, I joined the Computer Science faculty at the University of Tennessee. My teaching responsibilities included computer architecture and computer networking. My research interests included the design and use of software that used the collective power of large groups of workstations. While at the University of Tennessee, I received a CAREER Award from the National Science Foundation.

6. In 1997, I joined the Electrical and Computer Engineering faculty at Virginia Tech. My teaching responsibilities have included the design of embedded systems, computer organization, computer architecture, a variety of programming courses, and parallel computing. I have been cited multiple times on the College of Engineering's Dean's List for teaching.

7. In addition to the activities, education, and professional experience listed above, I have been involved in research projects that contribute to my expertise relating to this declaration. While at Virginia Tech, I have been a primary or co-investigator on government and industrial research grants and contracts in excess of five million dollars.

8. Many of the research contracts undertaken in the laboratory have involved collaboration and coordination with other groups to build a larger system. My responsibilities under the SLAAC project (a collaborative effort funded by the Defense Advanced Research Projects Agency involving the University of Southern

California, Sandia National Laboratory, Los Alamos National Laboratory, Brigham Young University, UCLA, Lockheed-Martin, and the Navy) included the development of a software system for monitoring, configuring, and controlling a networked collection of computers hosting specialized computer hardware. As part of the DSN project (a collaborative effort funded by the Defense Advanced Research Projects Agency involving UCLA and USC), I was responsible for designing algorithms and software for controlling and monitoring a large network of autonomous computer sensor nodes. This software was integrated with software from several other teams around the country for a set of field demonstrations over a three-year period.

9. In the Air Force-funded TEAMDEC project, I developed a web-based system to assist with decision-making by a geographically distributed team. Users of the system operated via browser-based clients (e.g., Microsoft Internet Explorer). The clients communicated with one another as well as a multi-tier server system. The system was made easier to use through user-recordable/editable scripts that allowed the team to quickly setup communications and share information such as web pages. The system used a database that allowed users to find the right script to accomplish their current task. In an earlier Air Force-funded project in the mid 1980's, I designed and implemented software for communication over MILNET,

portion of the ARPANET (the predecessor to the Internet) that was used for unclassified communication by the Department of Defense.

10. Another aspect of my work has involved computer security. As an example, one project included designing and implementing a computer architecture that protects the programs (and data) on the system from being reverse-engineered.

11. Other projects have involved the close coupling of computer hardware and software, including the writing of device drivers and simple operating systems, the design of hardware circuits, the design of new system architectures integrating low power data storage, architectures for secure computing, the modification of complex operating systems, and software for mediating between complex software packages. My work in e-textiles has focused on new architectures that integrate fault tolerant networks. I have designed image transmission systems for reliably transmitting images over wireless links using compression and error-correction techniques.

12. A detailed record of my professional qualifications is set forth in the attached Exhibit 1, which is my curriculum vitae, including a list of publications, awards, research grants, and professional activities

13. I have reviewed the following materials in connection with the preparation of this declaration:

- U.S. Patent No. 6,415,316 (the "'316 patent");

- The Petition and exhibits, including • U.S. Patents Nos. 6,233,600 ("Salas"), 5,729,734 ("Parker"), 5,933,811 ("Angles"); the Tittel reference and his declaration;
- The Patent Owner's Preliminary Response;
- The Institution Decision;
- A transcript of the deposition of Mr. Tittel; and
- Any other documents cited or referenced in this declaration.

My Opinions

The PTAB's Claim Constructions

14. For purposes of the Institution Decision, the PTAB construed the claim terms listed below as follows:

Term	PTAB Construction
"cohesive diary page" (claims 1 and 17)	"a page in which the content data and the page design are fully integrated for display."
"configuration information" (claims 1 and 17)	"information that determines what information will be displayed to a user who is viewing a cohesive diary page."
"privacy level information" (claims 1 and 17)	"configuration information that describes or specifies at least one user or category of users permitted to view particular content on a cohesive diary page."

Construction of Privacy Level Information

15. I agree with the Patent Owner's proposed construction of "privacy level information":

> "configuration information that describes or specifies the universe of one or more users or categories of users permitted to view particular content on a cohesive diary page"

16. As those of skill in the art would understand from the '316 patent's specification, the "privacy levels" of the '316 patent involve levels of access, where each higher level of access is granted at least all of the permissions of the lower levels of access. As the specification makes clear, a privacy level describes the set or universe of users who may view particular content – i.e., any user who has at least that level (or higher) may view the content, but users who do not have at least that level may not. That set of users can include just one person. In fact, the Specification expressly describes an "owner" privacy level depicted as a single profile head in Figure 4(e) disclosing a privacy level with a single member: the owner himself.

17. The disclosures in the '316 specification simply make express what is implied by the claim term "level." During his recent deposition, Mr. Tittel explained that the term level itself implies a "gradation." (Tittel Dep. at 9:17-21.) I agree. And as the specification makes clear, each gradation is a different group

of people, each grade more restrictive than the next in terms of who can view particular diary content.

Appropriate Level of Skill in the Art

18. I understand that an assessment of patent claims and prior should be undertaken from the perspective of a person of ordinary skill in the art as of the earliest claimed priority date for the patent, here, September 1, 1998.

19. I have reviewed Mr. Tittel's opinion on the appropriate level of skill in the art for the claims at issue and I agree with him. For convenience I have repeated the agreed upon level of skill below:

> A person of ordinary skill in the art as of September 1998 would possess at least a bachelor's degree in information technology or computer science (or equivalent degree or experience) with at least two years of practical experience in computer programming and development and implementation of network-based systems (such as, for example, systems for communicating over the Internet or the Web). Alternatively, a person having no such formal degree could qualify as a person having ordinary skill in the art, provided that he or she had at least four years of practical experience in computer programming and the development and implementation of network based systems (such as, for example, systems for communicating over the Internet or the Web).

20. I have used this definition in my analysis for this proceeding.

The Salas Files Cannot Be the Content Data of the Claims

21. Representative claim 1 of the '316 patent, along with the various proposed constructions of the claim limitation "privacy level information" appear below:

> What is claimed is:
>
> **1**. A method of organizing information for display, comprising:
>
> sending from a diary server to a user system, a diary program capable of being executed by a browser in the user system;
>
> 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;
>
> 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;
>
> 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
>
> sending, by the diary server to the user system, new diary information for changing the cohesive diary page.

Facebook's Proposal: "configuration information that describes or specifies at least one user permitted to view

particular content on a cohesive diary page" (Petition at 24.)

Patent Owner's Initial Proposal: "configuration information that describes or specifies which user(s) or categories of users are permitted to view particular content data on a cohesive diary page" (Initial Response at 12-13.)

Board's Institution Decision: "configuration information that describes or specifies at least one user or category of users permitted to view particular content on a cohesive diary page.'" (Decision at 10.)

Patent Owner's Current Proposal: "configuration information that describes or specifies the universe of one or more users or categories of users permitted to view particular content on a cohesive diary page" (Patent Owner Response at 50.)

22. The claim language itself, along with all the proposed constructions, make clear that the claimed content must be combined with the page design to create the cohesive diary page. I agree with Mr. Tittel that this is evident from both "the sending clause as well as in the assembling clause." (Tittel Dep. 12:4–13:3.)

23. Given this, the files themselves in Salas cannot be the claimed content because the files themselves are not combined with page design to form the eRoom page. Instead, the files are opened and displayed by the appropriate file application,

such as Microsoft Word. (*See* Salas at 2:41-43 ("A file request is received from a client workstation. An application capable of viewing the file is invoked."); 12:47-66.)

Neither the Salas File Access Information Nor the Other File Metadata Is Privacy Level Information

24. Because the Salas file access information pertains to the files themselves and does not concern whether or not a user can view any content data that is combined with page design information and displayed on the page, the Salas file access information is not the privacy level information required by the claims.

25. The other file metadata disclosed in Salas such as the filename, creation date, modified date, and which application should be used to open and edit the file is not privacy level information. This metadata has nothing do with privacy controls. It does not "describe or specify" one or more users who are "permitted to view particular content on a cohesive diary page" and thus it cannot be the claimed "privacy level information" either.

eRoom Access as Taught by Salas

26. Salas teaches a specific procedure to gain access to an eRoom in order to achieve its stated goal that an eRoom be private and secure:

> 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".

(Salas 4:66-5:6.)

27. Permission to enter the eRoom itself is checked by comparing the user's name or "authentication context" in a database table to determine if the user is on the list of members of that particular eRoom. (Salas 14:37:54.) The authentication context (e.g., username and password) is used to establish the identity of a remote user to the server. (Salas 14:11-16.)

28. In Salas, the files and data objects are requested by clients on the Internet from the server via the HTTP standard protocol. (Salas 13:59-61.) As described in Figure 5 and the associated discussion in the specification, the information (data objects) required to build and render an eRoom page is not returned in a single HTTP response from the server. Further, in addition to the information for building and displaying the eRoom page, external files are separately downloaded in Salas via HTTP. The client has to retrieve this information by making separate HTTP requests over the Internet to the server for data objects (e.g., synchronization requests) and for files (request to download a file). For every HTTP

query, the server has to ensure that the client has the appropriate access rights for the requested data object(s) or a file.[1]

29. The reason that this check for access rights must occur on every HTTP query is that the server must ensure (as cited above) that it only provides the files and data objects to clients with the appropriate access rights. Given that Salas is using the ubiquitous HTTP standard protocol on the WWW, any computer on the Internet could issue such a query and receive the file or data object from the server unless the server checks the user's credentials.[2]

30. This eRoom access control scheme is well-suited for a collaborative environment in which members of a team are working on a project. It is a virtual room in which team members are working together. There is no "public" in the

---

[1] The client does not receive any data objects or files for which it does not have the appropriate access rights. (Salas 11:54-58, 12:36-39, 14:33-36.)

[2] Salas indicates that the transfer of information from the server to the client via HTTP is traditional. Salas teaches the details of uploading files to the server, including authenticating the identity of the user making the request, because Salas indicates that sending files in this direction via HTTP is less common. (Salas 13:59-14:29.)

eRoom and no need to limit which team members can view information in the room because, just as in a real office, if there is a need for a room that is limited to just a subset of the team, an additional eRoom can be created for that sub-team. This is quite different from the diary of the '316 in which there is a single owner of the diary and that owner is controlling which information they would like to share with various types of people, including the entire public.

In Salas, Decisions Regarding Access Control Are Made at the Server, Not the Client, and the Access Control Information is Stored at the Server

31. Salas describes the storage of access controls in the database of the server. These access controls are described as a type of "data object." (Salas 3:49-51.) Access controls in Salas (whether for the eRoom page itself or for the linked-to files) are not only stored at the server, but also used there, as Salas makes clear, to check access rights in various scenarios including synchronization, setting and releasing edit locks, creating files, and modifying them.

Salas Synchronization Disclosures

32. Salas teaches that the Page Builder builds eRoom pages using information from the client's local database. The information in the client's local

database is provided to the client from the server via the synchronization process described in Salas.[3] (Salas Figure 5.)

33. Column 11 explains that the synchronization is done "to ensure that the objects from the local database and locally stored files are not stale before inserting them into the eRoom template." (11:14-16.) In operation, when an "object needs to be synchronized, the updated object may be requested from the server[.]" (11:32-33.) And in response to this request from the client workstation:

> The server 14 transmits to the client workstation 12' any data objects ***to which the user has appropriate access rights*** that also have a modification tag value greater than the modification tag value sent with the synchronization request.

(11:52-54.) Thus, the synchronization process ensures that the client has updated versions for all of the data objects and files the user has the right to access.[4] (Salas 11:54-58.) The client is not entrusted with data objects that it does not have the

---

[3] An exception to this statement is the data associated with the "unread" indicator.

[4] Again, the client does not receive any data objects or files for which it does not have the appropriate access rights. (Salas 11:54-58, 12:36-39, 14:33-36.)

right to access.

34. Because the server only sends objects to which the user has appropriate access rights, the access checking must necessarily be done at the server. I thus agree with Mr. Tittel on this point. He is correct that in Salas the server makes the judgment as to whether the client workstation has appropriate access rights. (Tittel Dep. at 44:2-22.)

35. In arguing that Salas sends file access control information from the server to the client workstations, the Petitioner relies on the end of the Salas synchronization discussion at the top of column 12, where Salas discloses that "local database metadata has been updated." (12:1-3) Specifically, the Petitioner suggests that this teaches updating *all* metadata including the access rights metadata:

> 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.)

(Petition at 40-41.) I disagree.

36. As an initial matter, the context of the synchronization discussion makes clear that the reference to updating "local database metadata" refers only to the modification tag, which is a critical part of the Salas synchronization technique,

as explained in column 11. The statement that "local database metadata has been updated" provides no indication of which specific metadata has been sent to the client workstation, whether it even includes file metadata, or the specific file metadata relevant to Petitioner's argument, namely, file access information.

37. The previous disclosure in column 11 unambiguously demonstrates that access rights are necessarily checked at the server *before* the objects are sent to ensure that the client only gets the information for which it has the appropriate access rights. This makes clear that whatever metadata is sent, it does not include access information.

38. I agree with Mr. Tittel that there is no discussion in words or in substance of updating access information file metadata in this passage:

> Q. … And then at the top of Column 12, it says, "Once synchronization has been accomplished and local database metadata has been updated," et cetera, et cetera; right?
>
> A. Yes.
>
> Q. And the updating of the local database metadata that they're talking about there is the updating of the modification tag; right?
>
> THE WITNESS: It -- it doesn't say that explicitly.

Q. But in context, that is what it's talking about, don't you think?

A. The preceding topic addressed both the new received modification tag value, but it also talks about the received data objects, and data objects have metadata associated with them. So it's not clear to me that the only thing that's changing is a modification tag count.

Q. Okay. But there's certainly no indication there that access information is being updated?

MR. WEINSTEIN: Object to the form.

Q. Isn't that right?

A. The words access information is being updated do not appear in that paragraph.

Q. And the substance of updating access information is not appearing there either; isn't that correct?

MR. WEINSTEIN: Object to the form.

THE WITNESS: There is no discussion of the updating of access information in that paragraph.

(Tittel Dep. at 49:24-51:7.) Given that there is no discussion of updating access information in that paragraph, in light of what is actually disclosed in that paragraph, in my opinion persons of skill in the art would not interpret updating

"local database metadata" as teaching sending of access information.

39. In fact, to interpret this "updating" passage as including the updating of access information would imply that access data was already present in the user system and in need of updating. But there is no suggestion in the synchronization discussions that, even though the access check takes place at the server, a duplicate of the access information is also (needlessly) stored at the user system. Nor is there any indication that the access rights change during the described synchronization process and would thus have any need to be updated.

Salas Edit-Lock Disclosures

40. The procedure illustrated in Figure 6 begins at the client workstations with the step (602) of checking "for local availability of file." The second Step (604) illustrated in the Figure—launching the "indicated application" for a downloaded file—is also performed by the client workstation.

41. Column 12 includes a detailed description of several steps that take place in between these two client steps of checking locally for the file (step 602) and launching the appropriate application after the file has been downloaded (step 604). These additional steps include an edit-lock check, an access-rights check, and, if appropriate, an edit-lock set. And they all take place at the server.

42. The first of these server-side steps is to check for an edit-lock after the first step at the client is complete:

> If the file is not present or is stale, then it must be downloaded from the server 14. The file is checked to determine whether another client workstation 12 has caused an edit lock to be set on the file indicating that the file is being edited. This may take the form of a database query for the object ID associated with the file which returns at least the metadata associated with the file indicating presence or absence of an edit lock.

(Salas 12:27-31.) Mr. Tittel testified at his deposition that this edit-lock check takes place at the server:

> Q. [T]here has to be a check to see whether there's an edit lock?
>
> A. Yes.
>
> Q. That's what it says?
>
> A. That's correct.
>
> Q. And that's at Column 12, lines 28 to 31?
>
> A. Correct.
>
> Q. And [then] it says, "This may take the form of a database query to the object ID associated with the file which returns at least the metadata associated with the file indicating presence or absence of an edit lock." Do you see that?

A. Yes.

Q. Now, what is your position on where that edit lock check is taking place? Is it at the local workstation or is it at the server?

A. It's at the server.

(Tittel Dep. at 54:9-55:1.) I agree with Mr. Tittel on this point.

43. Immediately after this edit-lock check, Salas states that "[i]f no edit lock has been set for a requested file, the access rights of the requesting user are checked." (Salas at 12:34-36.) This check, too, takes place at server and not the client.

44. Next, Salas discloses that when an edit lock is required because the user has the appropriate access rights and has indicated editing will occur, that "edit lock is set before the file is downloaded[.]" (Salas at 12:40-41.) This edit lock is set at the server.

45. Salas discloses that *after* these server-side actions are taken, the file is downloaded to the client and the second client-side step (604) of launching the appropriate application occurs at the client with respect to the downloaded file. (Salas at 12:47-49.)

46. Salas explains that the client workstation knows which application to launch in order to open and view the file (step 604) in one of three ways: (1) using

what's called the OLE standard;[5] (2) considering the file suffix; or (3) reading the "file metadata which is transmitted together with the file and indicates which application should be used to open and edit the file." (Salas at 12:49-55.)

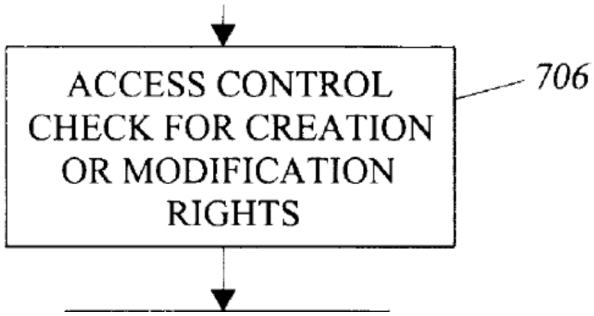
47. The third option uses "file metadata" sent to the client from the server. (12:49-55.) The file metadata being sent here is not "access information" but instead information that "indicates which application should be used to open and edit the file." Salas does not teach sending access information to the client here.

48. In sum, Salas teaches an edit-lock technique where access rights information is stored and used at the server, and never sent to the client. In fact, the server is the correct location to enforce this lock because it is a condition that must be enforced for all client computers, something only the server is in a position to enforce.

[5] The Object Linking and Embedding (OLE) is a Microsoft-developed standard. In this context, the use of OLE is referring to its capability for allowing a document to specify its type, where that type is based upon the application that created it. This type can be used to determine which application is used to view and edit the document.

Creating and Modifying Files

49. The Salas description of how eRoom users create or modify a file begins at column 13, line 10, by introducing the discussion of Figure 7 and the "intuitive drag and drop method" to upload new or modified files from the client to server. By dragging and dropping a new or modified file on the eRoom page, a client-side program (ActiveX control or background daemon) is activated to manage the upload process and the steps that it takes in this process are identified in Figure 7. The Salas disclosure here makes clear that the third step taken by that client-side program is to check the access rights of the user and that this is done over the network:



* * * * *

> The user's access rights are checked to ensure that the user possesses "create" or "modification" rights for the page to which the user desires to upload the file (step 706) and the file to be uploaded is stored to local mass storage (step 708). Access rights may be checked *over the network* in many ways.

(Salas Figure 7 and 13:27-32.)

50. This disclosure demonstrates that the access information is not on the client workstation because the server never sends them there. If this information were sent to the client and resident there, this "over the network" disclosure would be not just superfluous, but wrong. Mr. Tittel seems to agree:

> Q. And then the next sentence, in the next paragraph, "The user's access rights are checked to ensure that the user possesses, create, or modification rights for the page to which the user desires to upload the file, step 706. And the file to be uploaded is stored to local mass storage, step 708." Do you see that?
>
> A. Yes, sir.
>
> Q. Okay. So there has to be now a check of the access rights; correct?
>
> A. Yes.
>
> Q. All right. And it says that, "The access rights may be checked over the network in many ways." Do you see that?
>
> A. Yes.
>
> Q. Why are the access rights being checked over the network?
>
> A. This is a distributed system. That means that it is a system that involves actors that are in different locations.

In a situation where information has to be resolved about what operations are allowed, we want to communicate from the client to the server.

So there has to be a shared view of what kinds of communications are going to be allowed.

Q. Right.

A. And, again, for reasons of concurrency control, network access controls typically reside in either some kind of a directory services database or in some kind of network account database. And that involves communication with some kind of a server across the network.

Q. Okay. So the user workstation -- well, let's put it this way: The ActiveX control or the background daemon, which is resident -- well, that's resident at the user workstation; correct?

A. Yes.

Q. Okay. And so it, namely the ActiveX control or the background daemon, has to go over the network to find out whether the appropriate access rights exist, and it does that by going over the network to the server which has that access control information; isn't that correct?

MR. WEINSTEIN: Object to the form.

THE WITNESS: Yes, that is correct.

(Tittel Dep. at 59:11-61:10.)

51. Mr. Tittel is correct to observe that "network access controls typically reside in either some kind of a directory services database or in some kind of network account database" and that because these controls reside on the server there must be "communication with some kind of a server across the network." The only reasonable result of this observation in the context of this Salas disclosure is that the access information is not on the client workstation because the server never sends it there.

52. Salas next explains what happens "[i]f the user has the appropriate rights" to create or modify the file. This Salas disclosure makes clear that the file metadata is created and stored at the server and not the client:

> If the user has the appropriate rights, then a command is sent to the server to create a new file (step 710). This step may be skipped if the user is modifying a file instead of 40 creating a new file. However, creation of a new file allows the server to provide a degree of fault tolerance and version control, if those features are desired. If the server has been instructed to create a new file, a new object is created containing metadata associated with the data file and the file 45 is transmitted to the server 14 using HTTP (step 712). If a file modification is occurring, the server 14 updates the metadata contained

> by the data object associated with the file and the file is transmitted to the server 14 using HTTP (step 712). The server 14 associates the uploaded file with 50 the newly-created data object.

(Salas at 13:38-51.)

53. In other words, in the context of creating and modifying files, Salas describes access controls for a file as being stored and updated on the server in the form of a data object that also may include other information regarding the file (metadata). The data object containing the metadata is created on the server and then associated with the uploaded file on the server. When the uploaded file is a file that is modified (as opposed to newly created), the server also updates the data object (e.g., writing the modification date). Salas only indicates that the modified or created file itself is uploaded to the server, but teaches that the associated data object is maintained on the server. For both creating and modifying, it is the server that associates the uploaded file with the metadata. Mr. Tittel agrees with this basic understanding of this disclosure. (Tittel Dep. at 63:20-65:3.)

54. In sum, what these disclosures make clear to one of ordinary skill in the art is that file access control information is stored and used at the server, and never sent to the client.

Petitioner's Mirroring Argument

55. The Petitioner suggests that just because some file metadata is sent from the server to the client, all file metadata must have been sent. And at his deposition, Mr. Tittel suggested that Salas disclosed database mirroring. I disagree with both the Petitioner and Mr. Tittel.

56. As an initial matter, even if one accepts Petitioner's assertion that the information associated with icons in Figure 4 must necessarily be read from the data object containing the file metadata of 13:52-57, it does not follow that all of this information listed in 13:52-57 is sent to the client for at least the following reasons.

57. First, sending the file access information is flatly contradicted by repeated Salas disclosures, as explained in detail above. Every single action described in Salas as being related to the access information happens at the server. The "over the network" disclosure alone proves that the Salas access information is not sent to or stored on the client workstations and thus that Petitioner's "mirroring" theory is wrong.

58. Second, there is no reason to send the file access information to the client workstations metadata given that it is created, stored, modified, and used at the server. Mr. Tittel's initial answer to this question was mirroring, which he said was required when a Salas user creates an object for the first time. But as noted above, the file creation and upload disclosure actually makes the opposite clear.

When creating a file in the Salas system the file metadata is created by the server, stored at the server, used by the server, and never set to the client workstation.

59. Third, "database mirroring" is a term of art. It is a technique in which an exact copy of a database is maintained in a second location to allow continued operation if the first database becomes unavailable (e.g., if the disk for the original version fails). Salas does not describe database mirroring either explicitly or implicitly. In fact, in addition to the disclosures discussed above, Salas discloses the opposite of mirroring in two additional disclosures.

60. At 4:31-36 Salas makes clear that the client has only a subset of the server's database, and thus the client's version of the database is not the same as the server's database. Salas also teaches that some tables may be stored on the client workstation but not the server. Salas discusses an "unread" table used to indicate things that have been modified since the user read them last:

> However, the database 20' stored on the client workstation 12' may contain tables ***which are not stored by the server database 20***. For example, a client workstation 12 may store in its database an "unread" table which indicates which objects have been modified since the user of the client workstation 12' has last accessed those objects. An unread table may include a member identification field and a modification tag

indicating the last modification date and time of an object.

(4:37-46, emphasis added.) These two counter-example makes clear not just that "mirroring" is not required by the Salas system but that the opposite of mirroring is expressly disclosed. Taken together, these disclosures make clear that in Salas the asymmetry in information storage is bi-directional: the server stores more information than the client and vice versa. That is not mirroring.

61. Fourth, as Mr. Tittel admitted (*see* Tittel Dep. at 38:21-40:1), there is no disclosure in Salas of storing file access control on the client. And I disagree with his attempts to read that disclosure into the Salas reference.

62. Mr. Tittel says that mirroring is required for proper "concurrency controls." Mr. Tittel is apparently referring to the fact that Salas teaches a collaborative system in which multiple users may use the same eRoom at one time and that some of those users may edit the page or the files referenced by the page. Mr. Tittel attempts to use the need for concurrency control to read into Salas a requirement that all of the information, particularly the access control information, must be duplicated in the client database. Salas, however, explicitly addresses the issues raised by Mr. Tittel in significant detail[6] – there is no need to read into Salas

[6] Salas addresses the issue of concurrent access by users at 11:12-19 and

an undisclosed mechanism. Salas specifically uses edit locks to ensure that a file may not be edited by more than one user at the same time; those edit locks are, as agreed by Mr. Tittel, stored and enforced at the server and there is no need or use for them at the client. These edit locks are not alleged by the Petitioner to be the privacy level information nor do they control any content that is integrated into an eRoom page. Salas also goes in depth into its synchronization mechanism that ensures that a client will not operate on old (aka "stale") information because that client will check with the server to ensure that it has the latest version of the information that will be used to build an eRoom page. As explained elsewhere in this declaration, during this synchronization process, the server only sends the information to which the client has the appropriate access rights.

Access Control as Taught by Parker

63. Parker teaches an administrative application for managing access controls for shared files and folders. Parker teaches the use of this administrative application in conjunction with traditional file services operating on traditional servers.

---

indicates that the solution is the synchronization of the data used by the Page Builder for the eRoom page. Note that there is no disclosure of any use of access controls in the Page Builder of Salas.

64. The administrative application of Parker is not, for example, teaching a web-based client server system. The administrative application of Parker instead is an application that is running on a computer in communication with a server via file service communication protocol, not via HTTP, the protocol used for web-based communication such as that taught in Salas.

65. The creation of the administrative application in Parker was motivated by a desire to make it simpler for a computer system administrator to understand and set the access controls for the file system. A problem identified by Parker is that an administrator can, in the prior art, easily determine the access control settings that are assigned to a user for any given sharepoint, but it is difficult to ascertain whether or not the particular user is blocked from even seeing the sharepoint due to the access control settings assigned to that user for the parent folder(s) for that sharepoint. Parker addresses this by presenting the administrator with the resolved effective access control settings for any given sharepoint (i.e., taking into account the access control settings for the parent folder(s) of each sharepoint).

66. Parker does not teach that decisions with respect to access control are made at the client. Instead, as described above, Parker teaches a traditional file service in which the decision regarding access control are made at the server based on information stored at the server.

67. Parker states that:

> [U]ser's access privileges may then be saved to a file, to be *consulted by the file service* ***to determine*** whether that user has sufficient access privileges to one of the ***server*** files or folders.

(Paker 2:12-15, emphasis added.) That "file service" clearly operates on the file server. It is the only such file service disclosed in the Parker reference. Thus, this disclosure makes clear that access control decisions are made at the server based on information stored at the server. Mr. Tittel agrees. (Tittel Dep. 74:13-75:7.)

The Salas/Parker Combination Fails to Disclose Sending Privacy Level Information Under Any Construction

68. The Salas/Parker combination would yield a system that fails to send "privacy level information" to "describe or specify" who can view content on the cohesive diary page.

69. As the claim language makes clear, the relevant "content" is only what is displayed on the diary page, namely the icons, file names, lock symbols, and the like. The underlying files represented by the icons and filenames are not "content" because they are not ever displayed on the diary page, as Mr. Tittel properly admitted. So "privacy level information" for the Salas/Parker combination must "describe or specify" who can view the icons, file names, and lock symbols but that is not the case.

70. Parker's technique of modifying the icons to indicate access rights to the underlying files does not say anything about who can view the icons, file names, or lock symbols. To the extent that these items are sent from the server to the user system they simply represent the sending of content.

71. Sending content does not "describe or specify" who can view the content, which is a privacy level requirement. Sending content in Parker, just like sending content in Salas, is a step taken *after* it is determined at the server (just like in Salas) whether the particular user has that permission.

72. Finally, in the Parker blank out scenario, I agree with Mr. Tittel that if the icon and filename are to be blanked out it would make no sense to send any information at all. (*See* Ex. 2013, Tittel Dep. at 79:22-80:3.)

73. As such, the combined Salas/Parker system fails to disclose or suggest sending "privacy level information" to "describe or specify" who can view content on the cohesive diary page.

At Most, the Salas/Parker Combination Teaches a System that Sends Individual Access Rights Information and Not Privacy Level Information

74. The combination of Salas and Parker is also deficient under a construction of "privacy level information" that accounts for the claim term "level." At most the combined Salas/Parker system would have icon modifications or blanking that has to do only with the access rights of an individual user and has

nothing to do with the universe of users that have access to any particular piece of content.

No Motivation to Combine Salas with Parker

75. Petitioner has indicated that one of ordinary skill in the art would combine Salas with Parker to provide an eRoom user in Salas with an indication of whether the user would be able to view a file that is associated an icon in the item box of the eRoom. I disagree. This motivation is based, at least, upon Petitioner's misunderstanding of how Salas operates.

76. Salas does not teach any role in which a member of an eRoom does not have the right to read all of the content in the eRoom. The preferred embodiment of Salas describes these three roles. Salas, of course, isn't necessarily limited to these three roles. However, not only does Salas not describe other roles, the access control mechanism and eRoom interface pages are specific to these roles because they are appropriate for the problem faced by Salas. The Salas system provides "a collaborative work environment over a network" (Salas 1:7-8) that facilitates "a two-way exchange of information and project data between team members" (Salas 1:59-60) so that team members can work together on a common project. (Salas 1:66-2:11.) In a collaborative work-based system like that, there is good reason to share everything and no good reasons to hide anything. Hypothetical eRoom member

roles such as partial readers would be both inconsistent with the problem of Salas and Salas's solution to that problem.

77. Salas assigns different roles to different members within an eRoom:

> In some embodiments, users may be divided into ***three separate groups***: coordinator; reader; and participant. In this embodiment, a coordinator can add members to the eRoom and may supersede any rights granted to users. A reader is someone who has access to the eRoom solely to view the content of that eRoom while a participant is a user that may access the eRoom and may edit the objects and files contained in the eRoom as well as upload new objects and files to the eRoom.

(Salas at 14:42-50, emphasis added.)

78. In this scheme, if a user is a "reader" in a particular eRoom, then that user can view everything in the eRoom, but not modify or add content. If a user is a "participant" then that user can view all content in the eRoom as well as add and modify content. If a user is a "coordinator" in a particular eRoom, then that user has the additional right to add new members to the room.[7] (Salas 3:49-57 and 14:37-54.)

[7] Salas also names an Observer role without description (Salas 3:55), but this would, based on its name, appear to be the same as a Reader role.

79. Those of skill would understand that the "other embodiments" implied by the opening phrase in the passage quoted above are other embodiments where all the eRoom users can *at least* read all the content, such as where the users are divided not into three groups but just two: coordinator and participant. Those of skill would not understand the "other embodiments" to imply embodiments were some of the users were not permitted to read some of the content because that would defeat the entire purpose of Salas.

80. The Petitioner seems to agree, having stated:

> Salas makes clear, therefore, that the system *allows the user to view the list of files in the project* but the user may be [sic, not be granted] certain types of access to the file.

(Petition at 43, emphasis added.) I agree with the Petitioner's suggestion here that Salas is all about *types* of eRoom user access and not about access *per se*. That is how one of skill would understand the Salas disclosures as I've noted above. And giving every eRoom member at least viewing access to all the project files simply makes good sense given the very purpose of the Salas system, as explained above.

81. For these reasons, the Petitioner is incorrect to argue that there is any motivation to combine Salas with Parker to modify (or hide) icons for files the user doesn't have the right to view. In other words, if the user can be in the eRoom (is a

member) then he can see all content in the eRoom and there is no reason to "grey out" or hide an icon.

82. Further, there would not even be a reason to indicate (via modifying an icon) that a user can only read but not write certain content because the user's role in the room will determine this right for all content in the room. But that would still leave a system missing privacy level information (under any construction) or sending it from the server to the client.

83. As explained above, Salas does not teach a system in which a user may view some, but not all content on an eRoom page. Further, Salas does not teach that a user can view some, but not all files associated with the icons (or file names) on an eRoom page. Salas does not teach any differential level of viewing content within an eRoom. There is no need for this in Salas because users can simply create a new eRoom with such documents and then provide access to that eRoom for only those users who should view that content. See the discussion elsewhere in this declaration of how access control in Salas operates. Given that, for the reasons explained above, those of skill in the art would not be motivated to change Salas to include the file access indications taught by Parker and thus the proposed combination is an improper hindsight reconstruction that fails to render the asserted claims obvious.

* * *

I declare under the penalty of perjury that the foregoing is true and correct.

Executed this 14th day of October 2014 in Blacksburg, Virginia.

Mark T. Jones, Ph.D.