

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

BOX, INC., and DROPBOX, INC.,
Petitioners

v.

TOPIA TECHNOLOGY, INC.,
Patent Owner

Case No. IPR2023-00427
U.S. Patent No. 9,143,561
Filing Date: November 10, 2008
Issue Date: September 22, 2015

Title: Architecture for Management of Digital Files Across Distributed Network

Case No. IPR2023-00433
U.S. Patent No. 10,067,942
Filing Date: September 21, 2015
Issue Date: September 4, 2018

Title: Architecture for Management of Digital Files Across Distributed Network

REPLY DECLARATION OF RAVIN BALAKRISHNAN, PH.D.

Table of Contents

I.	INTRODUCTION AND QUALIFICATIONS	4
A.	Qualifications and Experience	4
B.	Materials Considered.....	4
II.	PERSON OF ORDINARY SKILL IN THE ART (“POSA”)	5
III.	STATEMENT OF LEGAL PRINCIPLES	6
IV.	OVERVIEW OF THE CHALLENGED PATENTS	6
V.	THE CHALLENGED CLAIMS OF THE ’942 AND ’561 PATENTS ALL RECITE SUBJECT MATTER THAT IS NOT DISCLOSED IN U.S. PROVISIONAL APPLICATION NO. 60/986,896	6
VI.	RESPONSE TO DR. SHENOY’S “BACKGROUND OF THE TECHNOLOGY”	7
VII.	THE CHALLENGED CLAIMS WOULD HAVE BEEN OBVIOUS IN VIEW OF SIGURDSSON IN COMBINATION WITH SHAPPELL.....	23
A.	Summary of the Prior Art	23
B.	Claims 1, 3, 4, 8, 10 and 11 of the ’561 Patent Would Have Been Obvious in View of Sigurdsson in Combination with Shappell.	24
C.	Claims 1, 3, 4, 10, 12 and 13 of the ’942 Patent Would Have Been Obvious in View of Sigurdsson in Combination with Shappell.	55
VIII.	THE CHALLENGED CLAIMS WOULD HAVE BEEN OBVIOUS IN VIEW OF BROWN IN COMBINATION WITH HESSELINK	87
A.	Summary of the Prior Art	87
B.	Claims 1, 3, 4, 8, 10 and 11 of the ’561 Patent Would Have Been Obvious in View of Brown in Combination with Hesselink.....	87

C.	Claims 1, 3, 4, 10, 12 and 13 of the '942 Patent Would Have Been Obvious in View of Brown in Combination with Hesselink.....	116
IX.	NO SECONDARY CONSIDERATIONS OF NON- OBVIOUSNESS.....	144
X.	ENABLEMENT AND REASONABLE EXPECTATION OF SUCCESS	144
XI.	CONCLUSION	145

Reply Declaration of Ravin Balakrishnan, Ph.D. in Support of
Petitions for *Inter Partes* Review of U.S. Patent Nos. 9,143,561 and 10,067,942

I, Ravin Balakrishnan, Ph.D., declare as follows:

I. INTRODUCTION AND QUALIFICATIONS

A. Qualifications and Experience

1. I previously summarized my qualifications and experience in paragraphs 1-13 of my Opening Declaration submitted in these matters (Ex. 1028 (hereafter, “Balakrishnan Open. Decl.”)), and I hereby incorporate by reference those paragraphs as if fully stated herein.

B. Materials Considered

2. The analysis that I provide in this Reply Declaration is based on my education, research, and experience, as well as the documents listed in § I.B of my Opening Declaration and cited, referenced, or discussed herein. In forming my opinions, I have read and considered, among other things, U.S. Patent Nos. 9,143,561 (“’561 patent”) [EX1001] and 10,067,942 (“’942 patent”) [EX1002] (collectively, the “Challenged Patents”) and the respective prosecution histories of the Challenged Patents. In addition to the documents listed in § I.B of my Opening Declaration, I have cited and relied upon the following documents in my analysis below:

Ex. No.	Description of Document
1040	Microsoft Computer Dictionary (5th ed.) (excerpt)
1041	RFC 2616, Hypertext Transfer Protocol – HTTP/1.1 (June 1999),

Ex. No.	Description of Document
	https://web.archive.org/web/20040614145003/https://www.ietf.org/rfc/rfc2616.txt (“IETF HTTP/1.1”)
1042	International Patent Application Publication No. WO 02/065359 A1 to Hepworth, et al. (“Hepworth”)
1043	U.S. Patent No. 5,416,842 to Aziz (“Aziz”)
1044	U.S. Patent No. 6,654,746 to Wong, et al. (“Wong”)

II. PERSON OF ORDINARY SKILL IN THE ART (“POSA”)

3. I hereby incorporate by reference § II of my Opening Declaration as if fully stated herein. I understand that Patent Owner’s expert, Dr. Prashant Shenoy, has provided the following opinions regarding the level of skill of a person of ordinary skill in the art (“POSA”):

It is my understanding that for purposes of this proceeding, the Petitioner is of the opinion that a POSA in November 2007 would have possessed a bachelor’s degree in computer science (or similar degree), and two years of work experience developing software, including at least some experience working with data storage and data synchronization. A person could also have qualified as a POSA with some combination of (1) more formal education (such as a master’s of science degree) and less technical experience, or (2) less formal education and more technical or professional experience in the fields listed above.

Based on my education, training, and professional experience in the field of the claimed invention, I exceed the level and abilities of a POSA at the time of the claimed invention. Although my qualifications may exceed those of one skilled in the art, my analysis and opinions regarding the Patents have been rendered from the perspective of one skilled in the art at the time of the invention.

Ex. 2025 (hereafter “Shenoy Decl.”), ¶¶ 15-16.

4. Based on the above testimony, it appears that Dr. Shenoy does not disagree with my characterization of the POSA. *See* Balakrishnan Open. Decl., ¶¶ 15-21. Thus, it is my understanding that Dr. Shenoy and I have applied the same definition of a POSA in our declarations.

III. STATEMENT OF LEGAL PRINCIPLES

5. I hereby incorporate by reference § III of my Opening Declaration as if fully stated herein, which details my understanding of relevant legal principles provided to me by counsel.

IV. OVERVIEW OF THE CHALLENGED PATENTS

6. I hereby incorporate by reference § IV of my Opening Declaration as if fully stated herein, which provides an overview of the Challenged Claims,¹ Challenged Patents, and file histories of the Challenged Patents.

V. THE CHALLENGED CLAIMS OF THE '942 AND '561 PATENTS ALL RECITE SUBJECT MATTER THAT IS NOT DISCLOSED IN U.S. PROVISIONAL APPLICATION NO. 60/986,896

7. I hereby incorporate by reference § V of my Opening Declaration as if fully stated herein, which explains the basis for my opinion that the Challenged

¹ The Challenged Claims are claims 1, 3, 4, 8, 10, and 11 of the '561 patent and claims 1, 3, 4, 10, 12, and 13 of the '942 patent.

Claims all recite subject matter that is not disclosed in U.S. Provisional Application No. 60/986,896, to which the Challenged Patents claim priority. Dr. Shenoy does not appear to dispute any of my analysis or conclusions in this section of my Opening Declaration. Therefore, there appears to be no dispute that the priority date of the Challenged Patents is November 10, 2008, which is the priority date I have applied in this Declaration and my Opening Declaration. *See* Balakrishnan Open. Decl., ¶ 15.

8. Dr. Shenoy vaguely references the “2007/2008 timeframe” in his Declaration as the relevant timeframe. *See, e.g.,* Shenoy Decl., ¶ 23. It is unclear why he references 2007 given that he does not appear to disagree with my conclusion that the Challenged Patents are not entitled to claim priority to their 2007 provisional application. *See* Balakrishnan Open. Decl., § V. In my opinion, the 2007 “timeframe” is irrelevant—the only relevant priority date is November 10, 2008.

VI. RESPONSE TO DR. SHENOY’S “BACKGROUND OF THE TECHNOLOGY”

9. In paragraphs 20-36 of his declaration, Dr. Shenoy provides opinions about the background of technology that is purportedly relevant to the Challenged Claims. Shenoy Decl., ¶¶ 20-36. I disagree with certain statements of Dr. Shenoy in this section of his declaration.

10. First, Dr. Shenoy vaguely states: “The ‘942 Patent^[2] discloses, a system to perform full-file synchronization across multiple local devices associated with a user through a server, e.g., a user centric architecture.” Shenoy Decl., ¶ 21. It is unclear what Dr. Shenoy means by “user centric” in this sentence. *Id.* With respect to the “user,” claims 1, 3, and 4 of the ‘561 patent only require the recited “first,” “second,” and “third” “electronic device[s]” to be “associated with a user” and that the user modify content of “first” and “second” “electronic file[s]” “of” the respective second and third electronic devices. ‘561 patent, claims 1, 3, 4. Similarly, claims 8, 10, and 11 of the ‘561 patent only require “first” and “second” “electronic device[s]” to be “associated with a user” and that the user modify content of “first” and “second” “electronic file[s]” “stored” on the respective first and second electronic devices. ‘561 patent, claims 8, 10, 11.

11. Similarly, with respect to the “user,” claims 1, 3, and 4 of the ‘942 patent only require “first,” “second,” and “third” “electronic device[s]” that are “associated with a user” and that “first” and “second” “electronic file[s]” be

² Although Dr. Shenoy references only the ‘942 patent in this portion of his Declaration, he notes in footnote 1 that he is also referring to the ‘561 patent, which shares the same specification. Shenoy Decl., ¶ 20 n.1. When discussing the ‘942 and ‘561 patents’ common specification, I also quote primarily from the ‘942 patent’s specification herein.

automatically received at the first electronic device from the second and third electronic devices, respectively, when the user modifies content of first and second electronic files. '942 patent, claims 1, 3, 4. Claims 10, 12, and 13 of the '942 patent only require “second” and “third” “electronic device[s]” that are “associated with a user” and “first” and “second” “electronic file[s]” that are automatically received at a “first” “electronic device” from the second and third electronic devices, respectively, when the user modifies content of the first and second electronic files. '942 patent, claims 10, 12, 13.

12. In my opinion, the aforementioned claim elements of the '561 and '942 patents that reference a “user” do not necessarily make the Challenged Claims “user centric” in every respect. Shenoy Decl., ¶ 21. First, in at least some of the Challenged Claims, not all of the recited “electronic device[s]” are required to be “associated with a user,” as noted above. Second, merely requiring certain electronic devices to be “associated with a user” does not necessarily render the entire systems or methods of the Challenged Claims “user centric”—the claimed “association” does not even necessarily mean that the user must *own* the devices, for example. Any association of the devices with a user suffices. Third, the fact that the Challenged Claims require a user to modify different electronic files stored on different devices for certain claims of the '561 patent merely requires the user to have access to such devices and make such modifications. This could be accomplished, for example, by

remotely accessing devices that are not physically near a user. As discussed in more detail below, certain of Dr. Shenoy's arguments appear to require more than mere association of certain devices and file modifications with a user. Because the claims do not recite and do not require such additions proposed by Dr. Shenoy, Dr. Shenoy is incorrect in adding such requirements.

13. Dr. Shenoy states that the Challenged Claims require a "server[.]" Shenoy Decl., ¶ 22. To the contrary, none of the Challenged Claims recites the word "server." See Balakrishnan Open. Decl., ¶¶ 63-64 (quoting Challenged Claims). The claims instead merely recite "electronic device[s]," which could be servers, laptops, desktop computers, mobile devices, etc. See *id.* The Challenged Patents state:

An embodiment treats storage like any other synch target, such as a desktop computer, or a cell phone. As such, any device owned or otherwise controlled by the user and running a synch application, such as synch application 380, as provided in an embodiment of the invention can perform the storage and/or synching functions described elsewhere herein. That is, the user device 280 or user device 210, rather than the server 230, may perform such functions.

'942 patent, 10:39-47. Thus, the Challenged Patents make it clear that a server is not required—any user computer (or any other computer) may be used to implement the synch application 380 shown in Figure 3 and described in the Challenged Patents.

14. Dr. Shenoy discusses what he and the Challenged Patents characterize as “various problems in connection with computerized file management systems[.]” Shenoy Decl., ¶¶ 23-26. Regardless of whether there were relative drawbacks of different prior art approaches to file synchronization and management in 2008, the Challenged Patents do not solve any problem or improve upon any of the existing approaches discussed by Dr. Shenoy or in the Challenged Patents. Instead, the Challenged Claims merely recite “automatically” moving files around using any of the many preexisting file synchronization technologies that already existed well before 2008.

15. Dr. Shenoy describes one of the supposed “problems” in prior art “[r]emote desktop software, distributed file systems, and file transfer protocol (FTP) systems” as the requirement that “files be[] stored in a single central location, such as the server, rather than locally on multiple devices of a user.” Shenoy Decl., ¶ 25 (citing ’942 patent, 1:63-2:46). However, Dr. Shenoy appears to argue later in his Declaration that at least some of the Challenged Claims **require** that files be stored at a single location that is distinct from multiple user devices using a client-server model (when, for example, a user’s device is offline and the file needs to be stored at a server, according to Dr. Shenoy). *See* Shenoy Decl., ¶ 128 (“However, updating designated subset of computers when data is updated as described in Hesselink, ¶183 is from the context of a file subscription discussed in Hesselink, FIGS. 11-13, which

is based on a peer-to-peer connection between two computers, as opposed to a client-network architecture.”). Therefore, this supposed “problem” in the prior art is actually required by the Challenged Claims, according to Dr. Shenoy. (I do not agree with Dr. Shenoy that a server is necessarily required to practice the Challenged Claims, as discussed above and in further detail below.)

16. I agree with Dr. Shenoy’s statement that “[t]here were many technologies for a client and a server to communicate with one another[,]” including “HyperText Transfer Protocol (HTTP) and HyperText Transfer Protocol Secure (HTTPS)[,]” which Dr. Shenoy describes as a “standard application-level protocol used by clients and servers to communicate over the World Wide Web.” Shenoy Decl., ¶ 27. Dr. Shenoy acknowledges that HTTP clients can use a “PUT” request to “send or upload files to a server using HTTP[,]” in addition to “GET” requests to request data from an HTTP server. *Id.* In addition, the exhibits cited by Dr. Shenoy note that clients may also use a “POST” command to “give information (e.g., annotation) to a server[.]” Ex. 2022, 654; Ex. 2023, 653. As Dr. Shenoy acknowledges: “In this manner, client and server can use HTTP to exchange files with one another.” Shenoy Decl., ¶ 27.

17. Dr. Shenoy opines that “the HTTP protocol is a request-response protocol, in which HTTP servers use the client-pull architecture, with the client-pull architecture being the one where a client (e.g., web browser) sends HTTP requests

to the server and the server sends HTTP responses after processing each request.”

Id., ¶ 28 (citing Ex. 1041,

<https://web.archive.org/web/20040614145003/https://www.ietf.org/rfc/rfc2616.txt>

(last visited January 30, 2024) (hereafter, “IETF HTTP/1.1”, § 1.4). However, none of the evidence cited by Dr. Shenoy supports his assertion that HTTP servers *always* use a “client-pull architecture.” Moreover, Dr. Shenoy neglects to mention that HTTP “requests” (such as “PUT” and “POST” requests) may contain data in the requests themselves that, when sent, effectively perform a “push” operation for the data communicated in such “requests.” For example, “Computer Networks” by Peterson and Davie explains that a client may use HTTP to perform a “write” operation to post data to a server. Ex. 2022, 653. Using the PUT or POST commands, an HTTP client can send data to a server, which is similar to a client-push architecture.

18. The IETF HTTP/1.1 reference cited by Dr. Shenoy states:

The HTTP protocol is a request/response protocol. A client sends a request to the server in the form of a request method, URI, and protocol version, followed by a MIME-like message containing request modifiers, client information, and possible body content over a connection with a server. The server responds with a status line, including the message’s protocol version and a success or error code, followed by a MIME-like message containing server information,

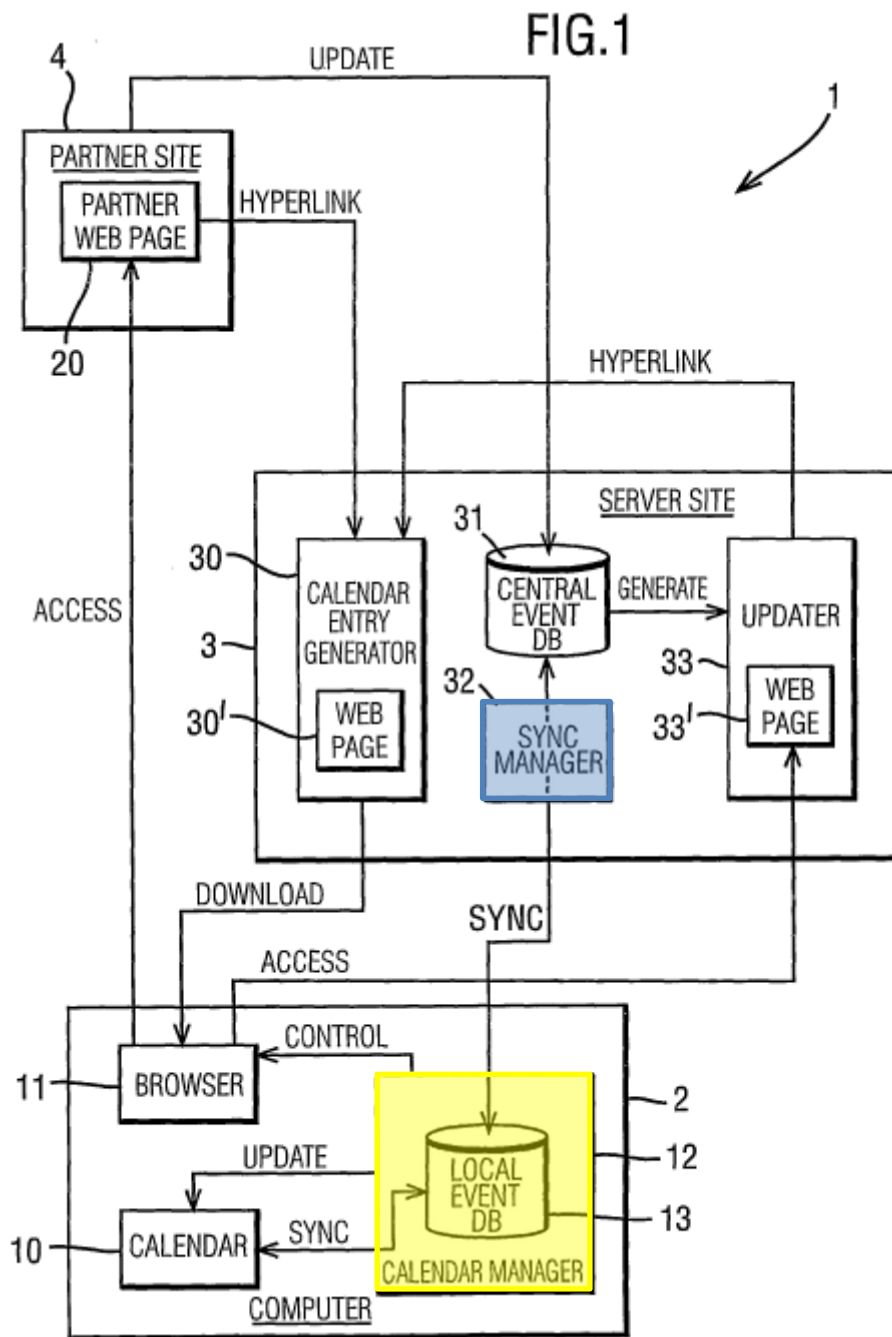
entity metainformation, and possible entity-body content.

IETF HTTP/1.1, § 1.4. The mere assertion that the HTTP protocol is a “request/response protocol” does not mean that all client interactions with an HTTP server must necessarily be of a “client-pull” nature. To the contrary, as noted above, the HTTP PUT and POST commands allow HTTP clients to send data to an HTTP server, and the server’s “response” may be used to determine whether the operation was successful or not. *See* Ex. 2022, 654 (“Like request messages, response messages begin with a single `START_LINE`. In this case, the line specifies the version of HTTP being used, a three-digit code indicating whether or not the request was successful, and a text string giving the reason for the response.”).

19. Dr. Shenoy states: “An HTTP server plays the role of a server in a client–server model by implementing the server part of the HTTP protocol” and asserts that “the HTTP server waits for incoming client requests and for each request it responds by providing the requested information or performing a requested task.” Shenoy Decl., ¶ 29. One of the “requested task[s],” as noted above, could be the provision of data *to the server from a client* via, e.g., the PUT or POST commands. Dr. Shenoy also fails to mention that many devices, such as those discussed in the Sigurdsson reference, may include *both* an HTTP client *and* an HTTP server, allowing the device to serve as both a client and a server, as discussed in more detail below.

20. It was well-known in the prior art to use HTTP to push updates from a server to a client. *See, e.g.*, Ex. 1042, International Patent Application Publication No. WO 02/065359 A1 to Hepworth, et al. (“Hepworth”), 26:13-14. For example, Hepworth describes an embodiment in which calendar events or updates may be “pushed” to a user’s digital calendar. *Id.*, 26:2-4. In one embodiment, “‘push’ protocols send the informational content to the user computer *automatically*, typically based on information pre-specified by the user.”³ *Id.*, 26:7-8. Hepworth explains: “This can be implemented . . . without user intervention. An example would be sending (‘pushing’) football scores as updates to fixture events in the calendar 10.” *Id.*, 26:8-11. The implementation described in Hepworth “uses HTTP to provide push-type functionality[.]” *Id.*, 26:13-14. To accomplish this, a “calendar manager” component in a user’s computer “makes use of [a] synchronisation process by which it automatically connects at intervals to [a] synchronisation manager 32 [at a server] . . . to effect addition of events from [a] server without the intervention of the user.” *Id.*, 26:15-18. This interaction is shown in Figure 1 below (with sync manager 32 highlighted in blue and calendar manager highlighted in yellow):

³ Emphasis added throughout, unless otherwise noted.



Id., Fig. 1.

21. In another embodiment, Hepworth describes “[c]ategory subscription” functionality that also uses the “pushed-events” functionality described above. *Id.*, 27:16-29:10. Hepworth explains: “Category subscription . . . enables a user to

subscribe to an entire category of ever-changing events based on the push functionality.” *Id.*, 27:19-21. Hepworth further explains:

One example of this would be a subscription to weather forecasts. Having done so, a user may thereby obtain a constantly-updating (i.e. ever up-to-date) forecast for a selected area. In order to enable forecast updating, the server site 3 receives at intervals during the day a “feed” of weather from a third party, e.g. a five day forecast for a number of global locations. ***Based on the user’s selected location, the server then regularly pushes this forecast as five different events - one event for each day of the forecast period - to the user. The server also effects a “push delete” to remove old forecasts.*** Consequently, at any given time, the subscriber to this service has the weather forecast for today and the following four days, in their calendar.

Id., 28:1-10. Hepworth further explains that among the data that may be transferred between a client and a server in its embodiments is an “XML file.” *See id.*, Abstract, 6:22-7:7, 12:11-12, 14:3-7, 14:21-23, 17:9-11, 18:14-19, 19:22-20:3, 21:5-20, 23:8-13. Hepworth describes this functionality without providing any detail as to how a POSA would need to implement it, making it clear that a POSA would have understood the implementation of this sort of functionality to be routine, well-understood, conventional, and within the background knowledge of a POSA. Specifically, in view of Hepworth (among other references describe herein and in my Opening Declaration), it would have been obvious to implement functionality

for pushing files from a server to a client and replacing older versions of files on the client with the new files.

22. As a more general matter, it was well-known, conventional, and obvious to send data via a “push” or “pull” model. In fact, the ’942 and ’561 patents themselves make this clear:

While a preferred embodiment of the invention has been illustrated and described, as noted above, many changes can be made without departing from the spirit and scope of the invention. For example, as an alternative to the approach described with reference to FIG. 3, wherein the transfer clients 310, 320 function to “push” modified or created files to the synch application 380, the synch application 380 may instead function to periodically “pull” or otherwise actively retrieve such files from the transfer clients 310, 320[.]

’942 patent, 10:48-56. Like Hepworth, the ’942 and ’561 patents merely reference “push” and “pull” models without further elaborating on how to implement either of them because a POSA would have viewed both of them as well-known, routine, and conventional design options that were obvious to try and trivial to implement.

23. The Challenged Patents refer to “a browser *or other applications*, running on one or more of the user devices 210, 280 [that] can cooperate in two-way communication with server 230 and one or more applications running on server 230.” ’942 patent, 7:51-54. The Challenged Patents provide no detail about how to implement any of these applications, confirming that doing so would have been

routine, obvious, well-known, and conventional prior to November 2008. Indeed, there were many well-known, conventional examples of applications used to push data from a server to a user's device that were in public use prior to November 2008. For example, EX1044, U.S. Patent No. 6,654,746 to Wong, et al. ("Wong") discloses an invention involving "the synchronization of the local copies of files on multiple mobile users' client computer hard disks to the current versions of the files on a network drive using only a single connection to the network drive." Wong, 1:19-24.

Wong further discusses:

Conventional products focus on delivering breaking news and information to user desktops automatically. Instead of the user constantly surfing multiple Web sites for stock quotes, news, weather, etc., conventional products aggregate and broadcast information automatically according to individual user preferences. Many companies incorporate "push" functionality into their products.

Id., 1:54-61. Wong further notes: "Microsoft and Netscape entered the fray with their own 'push' clients[,] IE 4.0 Active Desktop and Communicator's Netcaster, respectively." *Id.*, 2:1-3. Wong additionally notes: "Microsoft and Marimba, among others, have recognized the importance and potential of 'Electronic Software Distribution' (ESL) or 'software push' as a way of addressing the need to seamlessly deliver software updates across the network with the goal of directly impacting the total cost of ownership." *Id.*, 2:18-23. Wong further discusses

“delivering changes or ‘updates’ to the same physical files” using push. *Id.*, 2:53-55. Finally, Wong teaches: “Symantec Mobile Update (SMU) according to the present invention is the first solution in the document delivery or ‘push’ category in which currently existing data like reports, spreadsheets, presentations, etc., are delivered to the desk-top ‘intelligently’.” *Id.*, 12:19-24.

24. As can be seen from these portions of Wong, it was routine, conventional, and well-known to use “push-based applications” as taught in Wong. Wong, 1:63. In fact, the patent application that issued as Wong was filed in May 1999, over *nine years* before the November 2008 priority date of the Challenged Patents. A POSA would have viewed push-based applications as an obvious alternative to using an HTTP/HTTPS client/server to synchronize files. For example, a POSA would have found it obvious to try to use a push-based application to implement file synchronization.

25. The Moromisato reference discussed in my Opening Declaration also confirms that sending data via either a push or pull model was well-known, conventional, and obvious before November 10, 2008. *See* Ex. 1018 (“Moromisato”), 1:67-2:9 (“In order to change the data, either a ‘push’ or a ‘pull’ model may be employed. In a ‘push’ model, a collaborator generates a data change request that is forwarded to each other collaborator. The incoming data change requests are then used by each collaborator to modify its local data copy. In a ‘pull’

model, one collaborator modifies local data and offers up such changes upon asynchronous request. Other collaborators issue requests, fetch and gather changes, and subsequently modify their local data copies to reflect such changes.”). Although Moromisato’s discussion about the push and pull models appears in the context of a peer-to-peer implementation (*see id.*, 1:65-67), a POSA would have understood in view of references like Hepworth that implementing either push or pull to send data to or from a server and to or from a client would have been obvious, for the reasons above.

26. In view of the above, I disagree with Dr. Shenoy’s statement that “Dr. Balakrishnan’s testimony that ‘HTTP protocol does allow . . . for [ads] to be done in response to a client request, but also to send [ads] directly to the client from a server’ stating that ads are sent automatically from the server without any client request is incorrect.” Shenoy Decl., ¶ 32 (quoting Ex. 2024, 63:20-24). As noted above, data, including ads, may be pushed to a user’s device directly from a server using something like the Hepworth embodiments discussed above. The claims do not require an implementation in which a server must send unsolicited information to a client without the client first pinging the server periodically for updates, as discussed above in Hepworth, and as discussed in more detail below. Dr. Shenoy discusses an example in the context of a web browser (*id.*), but neither the Challenged Claims nor the prior art references relied upon in my Opening Declaration or herein are limited

to a web browser implementation.

27. Dr. Shenoy appears to incorrectly limit the claims to what he characterizes as the “standard HTTP protocol practiced during the 2007/2008 timeframe[,]” arguing that it “did not support push-based functionality where the server could ‘automatically’ send data to a client without the client making requests for data.” *Id.*, ¶ 34. The claims are not limited to a push-type implementation and are not limited to using HTTP. Moreover, as discussed above, Dr. Shenoy is incorrect in his overly narrow interpretation of HTTP that data could not be pushed from a server to a user’s computer before November 10, 2008.

28. A POSA would not have adopted Dr. Shenoy’s interpretation of the claims as requiring a server capable of pushing data to a client without any prior request from the client. For example, a POSA would have known that a user device connecting to a server from a public network using DHCP—a common networking protocol in use at the time—would have been assigned an IP address that would not have been initially recognizable to the server and could not have been associated with the user’s device absent some initial communication from the user’s device to the server. *See* Ex. 1043, U.S. Patent No. 5,416,842 to Aziz (“Aziz”), 10:57-61 (“[I]f the user is carrying a portable computer and connects to a public network facility where the computer is dynamically assigned an IP address using a protocol such as the Dynamic Host Configuration Protocol (DHCP), the user’s IP address is not

known”). Thus, a POSA would have understood the claims to encompass clients—especially clients coming back online after being offline—making the first request or contact to a server prior to a server automatically sending data to a client.

VII. THE CHALLENGED CLAIMS WOULD HAVE BEEN OBVIOUS IN VIEW OF SIGURDSSON IN COMBINATION WITH SHAPPELL

29. After reviewing Patent Owner’s Responses in the respective IPRs and Dr. Shenoy’s Declaration in support of Patent Owners’ Responses, it is still my opinion that (1) all of the Challenged Claims would have been obvious to a POSA in view of Sigurdsson in combination with Shappell and (2) all of the Challenged Claims would have been obvious to a POSA in view of Brown in combination with Hesselink. This opinion is based upon my review of the Challenged Patents and their prosecution histories, all other evidence cited and relied upon in this Declaration and my Opening Declaration, and my understanding of the knowledge and skill that would have been possessed by a POSA.

A. Summary of the Prior Art

30. I hereby incorporate by reference § VI.A of my Opening Declaration as if fully stated herein, which summarizes, *inter alia*, the Sigurdsson (Ex. 1016) and Shappell (Ex. 1020) prior art references.

B. Claims 1, 3, 4, 8, 10 and 11 of the '561 Patent Would Have Been Obvious in View of Sigurdsson in Combination with Shappell.

1. Independent Claim 1

- a. “the first application [executed on the first electronic device] is further configured to automatically transfer the modified first electronic file copy to the third electronic device”**

31. Dr. Shenoy opines that Sigurdsson does not disclose that “the first application [executed on the first electronic device] is further configured to automatically transfer the modified first electronic file copy to the third electronic device,” as this limitation appears in claim 1 of the '561 patent (similar limitations appear in each of the Challenged Claims). Shenoy Decl., ¶ 39. I disagree. In my opinion, Sigurdsson expressly discloses this claim element. *See, e.g.,* Sigurdsson, ¶ 26 (“Exchange of content between the computer devices can occur automatically without a user specifying that the content should be sent to another computer device.”). In addition, even if Sigurdsson does not disclose this claim element, it at least suggests or renders the element obvious, as discussed below. In addition, this element is rendered obvious in view of Sigurdsson in combination with Shappell, for the reasons discussed below and in my Opening Declaration.

32. Dr. Shenoy argues that this claim element requires the server to automatically transfer a modified file “without a request from the client for the modified file[.]” Shenoy Decl., ¶ 39. I disagree. Nothing in the claims or

specification requires transfer of the modified file from the server to the client without a request from the client for the modified file. In fact, the Challenged Patents reference “push” and “pull” approaches as if they are trivially interchangeable, and a POSA would understand from reading such disclosure that the claims do not require a server to push the modified first electronic file copy from the first electronic device to the third electronic device:

While a preferred embodiment of the invention has been illustrated and described, as noted above, many changes can be made without departing from the spirit and scope of the invention. For example, as an alternative to the approach described with reference to FIG. 3, wherein the transfer clients 310, 320 function to “push” modified or created files to the synch application 380, the synch application 380 may instead function to periodically “pull” or otherwise actively retrieve such files from the transfer clients 310, 320[.] Instead, the invention should be determined entirely by reference to the claims that follow.

’942 patent, 10:48-58. The ’942 and ’561 patents’ inventors knew how to distinguish between “push” and “pull” data transfer models based on the above disclosures but nevertheless decided not to limit the claims of the patents to require use of either one of these models by not using the word “push” (or the concept itself) in any of the Challenged Claims. Thus, Dr. Shenoy is wrong to simply assume that a push model is required, particularly given that he cites no language from the

Challenged Patents requiring such an interpretation.

33. One part of the '942 and '561 patents' specification states:

Typically, the server 230 will store one or more of the received files to the storage 270 only if the destination user device is offline or otherwise temporarily not in communication with the server 230. Upon resuming communication with the destination user device, the server 230 will transfer the temporarily stored files to the destination device.

'942 patent, 8:26-31. In my opinion, a POSA would understand this disclosure to contemplate a previously-offline user device first requesting to resume communication with the server before the server sends a modified file to the user device. Thus, Dr. Shenoy is wrong in assuming the claims require the absence of a request from a user device before a server sends a modified file to the user device.

34. The Challenged Patents expressly contemplate user devices connecting to a server via the internet. '942 patent, 7:28-32. As such, a POSA would have understood the Challenged Claims to cover the situation in which a previously-offline user device reconnects to the server via a public network over the internet using DHCP. For the reasons discussed above in § VI, the user device would have been **required** in this scenario to first reestablish communication with the server (not the other way around). Thus, Dr. Shenoy is incorrect in interpreting the Challenged Claims to require the **server** to transfer a modified file to a “third electronic device” “without a request from the client[.]” Shenoy Decl., ¶ 39.

35. Dr. Shenoy incorrectly states that “Sigurdsson describes that its system is implemented as a HTTPS client/server operating based on request/response process.” *Id.*, ¶ 40. To the contrary, Sigurdsson expressly describes that the client user devices described in Sigurdsson *and* the server may *each individually* include an HTTPS client *and* an HTTPS server. This is shown, for example, in Fig. 3 of Sigurdsson, which depicts “Client B” as including both an HTTPS client and server that bidirectionally communicates with a “Server” that a POSA would understand to include a reciprocal HTTPS server and client. *See also* Sigurdsson, Fig. 3, ¶ 47 (“The HTTPS Client/server 326 *can be used as a client* that tracks web access and can contain a Time Stamp 328 which associates viewed web pages with a time that they were accessed by the user. The HTTPS Client/server *can also be used as a server* to provide content to peer clients, which is described in more detail below.”). Sigurdsson further explains that all of its clients are fungible (i.e., they contain the same components), and Dr. Shenoy does not dispute this point. *E.g.*, Balakrishnan Open. Decl., ¶ 122. Thus, a POSA reading Sigurdsson would understand that clients can contain both HTTPS clients and servers that must necessarily communicate with a reciprocal HTTPS client or server on, e.g., Server 106 depicted in Fig. 3 of Sigurdsson.

36. Furthermore, Sigurdsson explains that “the HTTPS Client/server” “can transmit an HTTP post command containing the Client X Content 410 to the Server

106.” Sigurdsson, ¶ 54. Because the server may communicate with each client bidirectionally, and because each client is expressly described in Sigurdsson as including both an HTTPS client *and* an HTTPS server, a POSA would have understood or at least found it obvious from these disclosures that the client HTTPS servers could be used by, e.g., “Server 106” to transfer a modified file to a client device.

37. Although Sigurdsson describes use of clients’ HTTPS servers in certain embodiments involving peer-to-peer communication between clients, Sigurdsson also states that, even in such peer-to-peer implementations, “[t]he Server 106 can still serve as a drop-box for temporarily (or permanently) storing Client X Content 410 originating from any [of] the Clients 102, 104 or 406.” *Id.*, ¶ 51. Thus, a POSA would understand that Server 106 could communicate with clients in the same manner that clients communicate with each other, namely, that the Server could send a modified file to a client.

38. In fact, this is precisely what Sigurdsson contemplates in the alternative embodiment of step 606: “the Client X Content 410 can be sent directly to the Server 106, thus bypassing the Client B 104. ***The Server 106 can then transmit the content to the remaining client peers, such as the Clients B and C.***” Sigurdsson, ¶ 68. As noted by the Board in its institution decision for the ’942 patent, “Sigurdsson . . . does not describe the transfer of a modified file with the alternative embodiment of

step 606 as resulting from any request from Client C.” IPR2023-00433, Paper 17 (’942 Patent Institution Decision) at 36-37; IPR2023-00427, Paper 17 (’561 Patent Institution Decision) at 34.

39. Dr. Shenoy is incorrect in stating that “the standard HTTP protocol practiced during the 2007/2008 timeframe did not support push-based functionality where the server could ‘automatically’ send data to a client without the client making requests for data.” Shenoy Decl., ¶ 40. To the contrary, as discussed above in § VI, given that Sigurdsson expressly describes its clients as including HTTPS servers (in addition to clients), a POSA would have understood and found it obvious to use Server 106 to push a modified file to the client.

40. The Challenged Claims and specification do not require the use of a “push” model of data transfer or the absence of intervening steps between when a modified file is received from one user device at Server 106 and when it is forwarded by Server 106 to another user device. For example, claim 4 of the ’561 patent expressly requires the “first application” (on, e.g., Server 106 in Sigurdsson) to “store the modified first electronic file copy to the memory device associated with the first electronic device when the third electronic device is not in communication with the first electronic device.” ’561 patent, claim 4. The Challenged Patents describe a “synch application” that “polls” client devices to determine whether they are in communication with the server running the synch application. ’942 patent,

8:50-54. Thus, claim 4 and the '561 patent's specification contemplates at least one intervening step that does not appear in claim 1 but is required before a file may be transferred to the third electronic device in claim 4—namely, determining whether the first electronic device is in communication with the third electronic device. Given that claim 4 depends (indirectly) from claim 1, it necessarily follows that claim 1 must include within its scope implementations that do not immediately forward modified files from one client device through a server to another client device but, instead, implement other steps in between, such as allowing a previously-offline client to reestablish communication with the server, request a list of missing content, and then request the missing content itself.

41. Dr. Shenoy opines: “A POSA, considering the [Sigurdsson] reference as a whole, would not have understood the discussion of the alternate embodiment of step 606 described in ¶68, and the alternate embodiment of step 620 described in ¶80 to be divorced from the accompanying steps in Fig. 6.” Shenoy Decl., ¶ 42. I understand from counsel that a claim is obvious if a POSA would have been motivated to combine teachings from various embodiments in a single prior art reference based on the teachings of those embodiments, the general knowledge present in the art, and common sense. I have also been informed by counsel that a POSA has ordinary creativity, and that there is no requirement that there be an actual or explicit teaching to combine two embodiments in a single prior art reference. Dr.

Shenoy does not appear to be applying these principles from the perspective of a POSA; instead, he is analyzing only the precise embodiment associated with Fig. 6 in Sigurdsson in isolation, which I understand is not the correct analysis.

42. A POSA would have read the steps associated with Fig. 6 in Sigurdsson in view of other disclosures in Sigurdsson. For example, Sigurdsson discloses:

The Server 106 can facilitate synchronization of content among the user's computers. When the Client A 102 (e.g., the user's home computer) connects to the Server 106, Client B Content 112 originating from the Client B 104 is automatically copied to the Client A 102, as indicated by arrow 116. In some implementations, copying occurs for the Client B Content 112 not already stored on the Client A 102. Similarly, whenever the Client B 104 (e.g., the user's work computer) connects to the Server 106, any missing Client A Content 108 is copied automatically to the Client B 104, as indicated by arrow 118.

Sigurdsson, ¶ 31. A POSA reading this disclosure would have found it obvious to automatically transfer a modified file from Client A to the Server 106 and then from Server 106 to Client C, upon detecting that Client C comes back online when reading the embodiment associated with Fig. 6, as the concept of automatically transferring files for purposes of synchronization across multiple user devices as soon as possible is clearly reflected in the above excerpt (as well as many others in Sigurdsson). I disagree with Dr. Shenoy's approach of analyzing the embodiment associated with Fig. 6 of Sigurdsson without considering other disclosures in Sigurdsson. In

analyzing paragraphs 68 and 80 in Sigurdsson, I have at all times considered the teachings of Sigurdsson as a whole—not solely the embodiment associated with Figure 6.

43. In paragraph 44 of his declaration, Dr. Shenoy implies that paragraphs 61-82 of Sigurdsson are part of a single sequence. Shenoy Decl., ¶ 44. I disagree—the paragraphs instead describe a set of alternative ideas that may be used interchangeably in a “mixed peer-to-peer and server architecture[.]” Sigurdsson, ¶ 61; *see also id.*, ¶ 62 (“The system 400 can enable peer-to-peer content sharing and synchronization using a combination of peer-to-peer and ‘drop box’ techniques.”). A POSA would have been motivated to use the ideas explained in connection with Fig. 6 of Sigurdsson interchangeably with any of the embodiments described in Sigurdsson when, for example, such interchangeable use would result in obvious benefits to users of the systems and methods described therein, as more fully discussed in my Opening Declaration.

44. Paragraph 61 of Sigurdsson explains that, like the inventions described in the ’942 and ’561 patents’ embodiments, “polling” may be used to determine whether a client is offline or online, which confirms that a POSA would not read the Challenged Claims to require the absence of any user request before an updated file is automatically sent to a client from a server. *See* Sigurdsson, ¶ 61 (“In one implementation, determination of whether a client is online or offline can be made

by monitoring polling by the client. If no *request* is received from a client, then the client is assumed to be offline. Similarly, the Server 106 can maintain a table of client IDs and whether each client is online or offline. When a client comes online, it can notify the Server 106 that it is online, and the Server 106 can update its table of clients with the online status.”).

45. In connection with the embodiment associated with Fig. 6, Sigurdsson first states that “a user at the Client A 102 may edit and save a document on his work computer.” Sigurdsson, ¶ 63. Thereafter, Fig. 6 includes a “step 606” that may be implemented in two different alternative ways: (1) Client A may directly send its content to another client (e.g., Client B), as described in paragraph 65 of Sigurdsson, or (2) Client A may send its content “directly to the server,” as described in paragraph 68 of Sigurdsson. Thereafter, “the Client X Content 410 can be sent directly to the Server 106, thus bypassing the Client B 104. ***The Server 106 can then transmit the content to the remaining client peers, such as the Clients B and C.***” Sigurdsson, ¶ 68.

46. Dr. Shenoy states: “While ¶68 states that the ‘Sever [sic] 106 can then transmit the content to the remaining client peers, such as Clients B and C’ it in no way teaches or suggests that Server 106 *automatically* transfers the content to a client.” Shenoy Decl., ¶ 46 (emphasis in original). I disagree. In my opinion, paragraph 68 of Sigurdsson *expressly* teaches automatically propagating content

received from Client A at Server 106 to another client, even under Dr. Shenoy's flawed understanding of what is required by the Challenged Claims. The paragraphs that follow paragraph 68 do not relate to the embodiment described in paragraph 68, in which content received from Client A is sent by Server 106 to Clients B and C, given that Client C is offline in the paragraphs that follow paragraph 68. Therefore, I disagree with Dr. Shenoy's statement that "the very next paragraph [after paragraph 68] begins the discussion of *how* Server 106 transfers the content, through steps 612 to 626, when Client C comes back online." *Id.* (emphasis in original).

47. I also disagree with Dr. Shenoy's statement that "[a] POSA would understand that Sigurdsson at ¶ 68 discloses that the alternative embodiment of step 606 is part of the overall sequence for handling synchronization *when a client is offline* when a transfer of content is to take place." *Id.*, ¶ 47 (emphasis in original). To the contrary, paragraph 68 expressly discussed the Server 106 sending content to Client C, which would, of course, be impossible if Client C were offline (as it is described in the paragraphs that follow paragraph 68).

48. I likewise disagree with Dr. Shenoy's statement that, "[c]onsidering the Sigurdsson reference as a whole, a POSA would have considered Sigurdsson's alternative embodiment of step 606 as limiting the transfer of the content to using step 620 and its associated steps 612-620 and 626-630." Shenoy Decl., ¶ 48. To the contrary, Fig. 6 includes a horizontal, dashed line that clearly separates the

alternative embodiment of step 606 described in paragraph 68 from the remaining steps shown in Fig. 6. This is because Client C is online, not offline, in paragraph 68's embodiment in which "the Client X Content 410 can be sent directly to the Server 106" and thereafter "[t]he Server 106 can then transmit the content to the remaining client peers, such as the Clients B and C." Sigurdsson, ¶ 68. Dr. Shenoy ignores this text in paragraph 68, pretending that, instead, paragraph 68 has something to do with the steps that follow paragraph 68 when Client C is offline. This is incorrect.

49. The alternative embodiment for step 606 described in paragraph 68 also contradicts Dr. Shenoy's incorrect statement that, "[i]f Client C was online, Sigurdsson teaches that the transfer would be performed using the peer-to-peer technique that does not involve transferring content from the server." Shenoy Decl., ¶ 48. This is clearly incorrect—paragraph 68 expressly contemplates Client C being **online** (otherwise, it could not receive content from Server 106) and expressly contemplates Client A content stored at Server 106 being sent to Clients B and C. Paragraph 68 also contradicts Dr. Shenoy's flawed assumption that all transfers to a client "would not be performed by the server using the 'drop box' technique" when a sending client and receiving client are online. Shenoy Decl., ¶ 48. To the contrary, paragraph 68 makes it clear that the peer-to-peer and drop-box techniques may be employed interchangeably when sending and receiving clients are online—the

sending client may either send to Server 106, which can then forward the file to the receiving client, or the sending client may send to the receiving client directly. A POSA would have viewed either of these options as obvious, conventional, and well-known design choices that would have been obvious to try in view of Sigurdsson's disclosures.

50. The alternative embodiment for step 606 described in paragraph 68 also contradicts Dr. Shenoy's incorrect statement that, "[w]hile it is true that Sigurdsson discloses that the Server 106 can transmit the content to the remaining client peers, such as the Clients B and C, Sigurdsson is clear that this transmission of the content by the Server 106 is performed *only* after receiving a request for the content from the Clients B and C." Shenoy Decl., ¶ 50 (emphasis in original). There is no such "request for the content from the Clients B and C" disclosed in paragraph 68, and Dr. Shenoy is wrong to read paragraph 68 as requiring the request from Client C discussed in the paragraphs that follow paragraph 68 in Sigurdsson.

51. I disagree with Dr. Shenoy's statement that "[a] person of ordinary skill in the art would not understand the statement in ¶80 [in Sigurdsson] to teach or suggest Server 106 automatically transferring the modified content to Client C." Shenoy Decl., ¶ 49. Paragraph 80 of Sigurdsson discloses: "In step 626, the server sends the missing content to the third client. For example, referring to FIG. 4, the Server 106 sends the Client X Content 410 to the Client C 406." Sigurdsson, ¶ 80.

In Fig. 6, step 626 is depicted as occurring after Client C comes back online (after being offline) and after Client C requests and receives a list of missing content and requests missing content from the server in step 626. Sigurdsson, ¶¶ 69-80, Fig. 6. Again, I read the disclosures in paragraphs 69-80 to be a recitation of interchangeable ideas and concepts in view of which the POSA would have found it obvious to automatically send content from Client A at Server 106 to Client C after Client C comes back online.

52. In addition, for the reasons discussed above, a POSA would understand that Server 106 automatically sends the Client A content to Client C, even though Client C requests a list of missing content and requests the missing Client A content from Server 106. A POSA would view the sending of Client A content from the server to Client C as described in paragraphs 69-80 to be automatic both from the server's perspective and the user's perspective. Even if Sigurdsson's systems and methods are only automated from the user's perspective, that is all that the Challenged Claims require in view of Dr. Shenoy's opinion that the Challenged Claims are "user-centric." Shenoy Decl., ¶ 22. The intervening steps do not change the automatic nature of sending the Client A content by the server to Client C. As noted above, all independent claims recite "comprising," which I understand to permit additional steps such as those involving Client C requesting a list of missing content and requesting the missing content from the server while still practicing the

Challenged Claims.

53. In paragraph 53 of his declaration, Dr. Shenoy presents a false dichotomy between “a first scenario when **clients are currently online** and a second scenario when the offline **client comes online again**” in which, in the first scenario (according to Dr. Shenoy), a peer-to-peer model must be used and in which, in the second scenario (according to Dr. Shenoy), the “drop box” model must be used. Shenoy Decl., ¶ 53 (emphasis in original). There is no disclosure in Sigurdsson that, when clients are online, they *must* use a peer-to-peer paradigm, whereas, when clients are offline, they *must* use the “drop box” paradigm. In fact, paragraph 68 of Sigurdsson is an express example to the contrary—it expressly discloses sending content received from Client A at Server 106 to Clients B and C (which obviously must be online to receive the content). Sigurdsson, ¶ 68. In paragraph 53, Dr. Shenoy again misrepresents Sigurdsson’s disclosure in paragraph 68 as requiring a request from a client before the server may send content to the client (Shenoy Decl., ¶ 53)—but there is no such disclosure in paragraph 68, and no other paragraph of Sigurdsson so limits paragraph 68.

54. Dr. Shenoy incorrectly asserts that “the first scenario in which the client is already online and Sigurdsson at ¶ 65 unambiguously states that the peer-to-peer technique should apply[.]” Shenoy Decl., ¶ 55. This sentence effectively ignores paragraph 68, which is expressly described in Sigurdsson as an “[a]lternative

embodiment of step 606” described in paragraph 65. The peer-to-peer embodiment in paragraph 65 is only one alternative of multiple, with paragraph 68 disclosing an alternative embodiment that uses Server 106. Paragraph 68 both expressly discloses and suggests to a POSA that the Server 106 automatically transfers content 410 to Client C 406 without receiving a request from Client C 406—no such request is referenced in paragraph 68 of Sigurdsson.

55. Dr. Shenoy again mischaracterizes Sigurdsson in his statement: “Therefore, in every case involving a Server 106 transmitting content to the Client C, Sigurdsson requires the Server 106 to receive a request from the Client C before the Server 106 transmits the content 106 to the Client C.” Shenoy Decl., ¶ 60. Again, the alternative embodiment of step 606 in paragraph 68 requires no “request from the Client C before the Server 106 transmits the content 106 to the Client C”—such a requirement simply does not appear in paragraph 68. In addition, as discussed below, even if a request were required from a client before the server sends content to the client, it is my opinion that Sigurdsson still discloses the limitation requiring automatic transfer of a modified file from the “first electronic device” to a “third electronic device” (as recited in claim 1 of the ’561 patent). For the foregoing reasons, and contrary to Dr. Shenoy’s assertions, a POSA would *not* have “consider[ed] Sigurdsson’s disclosure to limit the transfer described for the alternative embodiment of step 606 to step 620 and its associated steps 612-618 and

626-630.” Shenoy Decl., ¶ 61.

56. Dr. Shenoy opines that, “even if Sigurdsson were read to have one client send the modified content to the server to synchronize with another client that was online at the time, this hypothetical example would fall outside the scope of claims 4 and 11.” Shenoy Decl., ¶ 61. As I explained in my Opening Declaration (¶¶ 543, 550), claims 4 and 11 of the ’561 patent would have been obvious in view of the combination of Sigurdsson’s disclosures in paragraphs 66 and 68. In particular, paragraph 68 states that Client A may send data “directly to the server” and paragraph 66 states: “in this case, *the third client is offline, so the content is transmitted to the server for temporary storage.*” Sigurdsson, ¶¶ 66, 68. Thus, Sigurdsson, clearly suggests and renders obvious the concept of temporarily storing content at the server when Client C is offline.

57. In addition, as I explained in my Opening Declaration, the limitation for using the “first application” on the first electronic device to “automatically transfer the modified first electronic file copy to the third electronic device” would have been obvious in view of other Sigurdsson disclosures. ’561 patent, claim 1. Sigurdsson’s Server 106 in Figure 4 includes a “Synchronization Manager” that “synchronize[s] content stored on and shared between” different clients (e.g., “Client A” and “Client C”). Sigurdsson, ¶ 48. As noted in my Opening Declaration, a POSA would understand that “synchronization” in this context means “version

comparisons of copies of . . . files to ensure they contain the same data.”

Balakrishnan Open. Decl., ¶ 522 (quoting Ex. 1021, Microsoft Computer Dictionary (5th ed. 2002), 506 (entry for “synchronization”).

58. The steps associated with Figure 6 must be read in the context of these disclosures related to Fig. 4. Sigurdsson, ¶ 61 (“**FIG. 6** is another sequence diagram showing an illustrative method for sharing and synchronizing content using a mixed peer-to-peer and server architecture of **FIG. 4** when a client that is offline comes online.”). In particular, the steps in Fig. 6 describe an automated process for (1) detecting when Client C comes back online, (2) passing a list containing metadata to enable Client C to see what content it is missing, (3) receiving a request from Client C for missing data, and (4) sending the missing data to Client C. Balakrishnan Open. Decl., ¶ 523. A POSA would have understood the steps in paragraphs 69-81 of Sigurdsson to be an automated process that includes sending Client A content stored at Server 106 to Client C when Client C comes back online. *See, e.g.*, Sigurdsson, ¶ 79 (“Additionally, the Client C 406 can request the content from the Server even if the one or more other clients are online. For instance, the Client C may request the content from the Server if the other clients have less available bandwidth relative to the Server.”).

59. As discussed above, the fact that Client C may send a request for the Client A content stored at the server before receiving it in paragraphs 69-81 does not

change the automated nature of the process described in these paragraphs. None of the steps in paragraphs 69-81 are described as being performed by the user or requiring user input (or input from anything else). For example, paragraph 69 states that Client C “requests a list of missing content from the Server 106 *when it connects to network that links the server and other clients, such as the Internet.*” Sigurdsson, ¶ 69. A POSA would understand that this is how a typical client using DHCP would connect to the server (via a request from the client to the server) for the reasons discussed above—e.g., the server would not know the client’s IP address when the client connects to a public network on the internet. Thus, paragraph 69 makes it clear that Client C’s request for a list of missing content is contemporaneous with Client C’s connection to the server—a POSA would understand that this functionality would necessarily be automated from both the user’s and the server’s perspectives.

60. Paragraphs 70-71 describe an automated process for determining the list of content that may be missing from Client C or that Client C may request using an automated “priority formula.” Sigurdsson, ¶¶ 70-71. Again, the POSA would understand these steps to be automated from both the server and user’s perspective—the client automatically determines what to request without the need for any outside input. Paragraphs 73-74 note that the list of missing content may specify content that Client C “can obtain from the server.” *Id.*, ¶¶ 73-74. In one alternative

embodiment of step 620, Client C “can obtain the missing content from the server.”

Id., ¶ 79. In paragraphs 80-81, the Client A content stored on Server 106 is sent to Client C from the server and stored at Client C. *Id.*, ¶¶ 65, 68, 80-81. A POSA would understand that all of these steps would be implemented in software and automated from both the server’s and user’s perspectives.

61. In connection with Fig. 1, Sigurdsson discloses:

The Server 106 can facilitate synchronization of content among the user’s computers. When the Client A 102 (e.g., the user’s home computer) connects to the Server 106, Client B Content 112 originating from the Client B 104 is ***automatically*** copied to the Client A 102, as indicated by arrow 116. In some implementations, copying occurs for the Client B Content 112 not already stored on the Client A 102. Similarly, whenever the Client B 104 (e.g., the user’s work computer) connects to the Server 106, any missing Client A Content 108 is copied ***automatically*** to the Client B 104, as indicated by arrow 118.

Sigurdsson, ¶ 31. Sigurdsson later explains that Fig. 2 is a “sequence diagram showing an illustrative method 200 for sharing and synchronizing content across multiple client devices ***according to the implementation shown in FIG. 1.***” *Id.*,

¶ 32. Thus, Fig. 2’s embodiment is automated like the embodiment in Fig. 1. Likewise, Sigurdsson explains that “FIG. 4 is a schematic showing a system 400 for sharing and synchronizing content” and that, in its depicted “system 400,” “content still can be shared among the Client A 102, the Client B 104 and a Client C 406

using the Server 106 ***in a method similar to that of method 200 of FIG. 2.***” *Id.*,

¶ 51. Thus, Fig. 4’s synchronization embodiments are also automated like those associated with Figs. 1 and 2. Finally, Sigurdsson states: “FIG. 6 is another sequence diagram showing an illustrative method for sharing and synchronizing content ***using a mixed peer-to-peer and server architecture of FIG. 4*** when a client that is offline comes online.” *Id.*, ¶ 61. Because the synchronization associated with Fig. 1 is described as being automated, and because the methods of synchronization associated with Figs. 2, 4, and 6 are described as operating similar to the synchronization associated with Fig. 1, a POSA would have understood Fig. 6’s embodiment (described in paragraphs 61-81) to be automated.

62. For the reasons discussed above, I disagree with Dr. Shenoy’s opinion that “Sigurdsson is directed to a pull mechanism, and fails to automatically transfer the modified first electronic file copy to the third electronic device associated with the user.” Shenoy Decl., ¶ 63. Sigurdsson discloses that its clients and Server 106 includes ***both*** HTTPS clients ***and*** HTTPS servers for bidirectional communication, and paragraph 68 expressly discloses pushing content from Server 106 to Clients B and C. In addition, a POSA would have understood how to implement push-type data transfers including by using disclosures such as those in the Hepworth reference, discussed above. No modification of Sigurdsson would be necessary to implement push functionality—it is already expressly recited in paragraph 68.

Server 106 is not described as an “HTTP server[,]” as Dr. Shenoy misleadingly suggests (*id.*). Instead, as discussed above, a POSA would have understood Server 106 as including both an HTTPS client and server for bidirectional communication with clients’ HTTPS clients and servers, including communications to clients from Server 106 to effectively push data/files to clients.

63. Dr. Shenoy’s own cited evidence confirms that a POSA would have understood Server 106 to implement push-based data/file transfers to clients based on Sigurdsson’s disclosure of its clients including HTTPS servers, because a POSA would know that HTTPS servers may receive PUT and POST messages to effectively push data to an HTTPS server. IETF HTTP/1.1, § 9.5 (“The POST method is used to request that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line.”); *id.*, § 9.6 (“The PUT method requests that the enclosed entity be stored under the supplied Request-URI.”).

64. Moreover, the Challenged Claims do not recite the word “push” and do not require a push mechanism to transfer data from the server to any client. The inventors knew how to use the words “push” and “pull” to describe data transfers and purposely did not limit the Challenged Claims to being implemented via push or pull.

65. In addition, neither Sigurdsson nor Shappell *requires* use of

HTTP/HTTPS, and a POSA would have known how to implement push-based file synchronization using push-based applications. The Challenged Patents refer to “a browser *or other applications*, running on one or more of the user devices 210, 280 [that] can cooperate in two-way communication with server 230 and one or more applications running on server 230.” ’942 patent, 7:51-54. The Challenged Patents provide no detail about how to implement any of these applications, confirming that doing so would have been routine, obvious, well-known, and conventional prior to November 2008. Indeed, there were many well-known, conventional examples of applications used to push data from a server to a user’s device that were in public use prior to November 2008, as discussed above in § VI with respect to the Wong reference.

66. Dr. Shenoy incorrectly states: “Sigurdsson at ¶ 31 does not correspond to the ‘automatic send’ limitation (*i.e.*, without a client request).” Shenoy Decl., ¶ 66. I disagree. Paragraph 31 clearly and unambiguously discloses an embodiment in which “Server 106 can facilitate synchronization of content among the user’s computers” by automatically sending updated or new content from one client through Server 106 to another client. Sigurdsson, ¶ 31. There is nothing in Sigurdsson that limits the embodiment in paragraph 31 to any “pull” approach associated with Fig. 2 as Dr. Shenoy suggests. Shenoy Decl., ¶ 66. Instead, Sigurdsson makes clear that Fig. 2 describes one possible implementation of an

embodiment using the system shown in Fig. 1, and that the client-request mechanism from Client B to the server, as shown in Fig. 2, is only one possible implementation of the automated synchronization process described in Fig. 1. Sigurdsson, ¶ 32. The association of Fig. 2 with Fig. 1's system, as discussed in paragraphs 31-32 of Sigurdsson, confirms that, even though Fig. 2's flow diagram includes a client request to the server, the flow diagram is still automated consistent with paragraph 31 and its description of Fig. 1.

67. There is nothing in the Challenged Patents or their file histories, such as a clear disclaimer or disavowal of claim scope or lexicography, that could justify limiting the claims to cover embodiments in which no client requests are sent to Sigurdsson's server. In fact, the Challenged Patents teach the opposite: a client *must* respond to a server's polling request to allow the server to first determine that the client is in communication with the server. '942 patent, 8:45-59. Even though the Challenged Claims do not expressly recite using such a client request, the Challenged Claims use the word "comprising," which means that Sigurdsson's embodiments disclose and render obvious the Challenged Claims even though they may contemplate certain automated communications between clients and the server that are not expressly recited in the Challenged Claims.

68. Dr. Shenoy incorrectly states: "Since Sigurdsson *expressly* states that Fig. 4 implements a method similar to Fig. 2, the method of Fig. 4 also includes steps

are performed by the Client B 104 to receive content.” Shenoy Decl., ¶ 74 (emphasis in original). To the contrary, Sigurdsson states: “In the system 400, content still *can be* shared among the Client A 102, the Client B 104 and a Client C 406 using the Server 106 in a method similar to that of method 200 of FIG. 2.” Sigurdsson, ¶ 51. The words “can be” are not the same as “must be.” Thus, the Client B request shown in Fig. 2 is not required to implement steps associated with synchronization methods that use Fig. 4’s system.

69. For the reasons above, the embodiments in paragraphs 31 and 68 of Sigurdsson make it clear that client requests are not necessary to implement the automatic sending of a modified file from the first electronic device to the third electronic device recited in claim 1 of the ’561 patent. Dr. Shenoy is incorrect in opining to the contrary. *See* Shenoy Decl., ¶ 75. Moreover, as discussed above, the Challenged Claims cover the use of both pull and push mechanisms, so Dr. Shenoy’s arguments that erroneously attempt to limit Sigurdsson’s embodiments to using pull mechanisms are ultimately irrelevant.

70. Even if the Board were to conclude that one or all of Sigurdsson’s embodiments are limited to pull-based HTTP mechanisms as Dr. Shenoy incorrectly asserts (*see* Shenoy Decl., ¶ 79), it would have been obvious in view of the embodiments disclosed in paragraphs 31 and 68 to push data from Server 106 to a client (as discussed in paragraph 47 of Sigurdsson). Use of such functionality would

have been in the background knowledge of the POSA, as demonstrated by the fact that, e.g., paragraphs 45 and 47 of Sigurdsson disclose the use of HTTP “post” messages without even describing what they are. *See* Sigurdsson, ¶¶ 45, 47. The embodiments in paragraphs 31 and 68 of Sigurdsson describe pushing file updates from Server 106 to clients without the need to go into any detail about HTTP or other well-known file transfer methods that could have been used to implement Sigurdsson’s systems and methods that implement such functionality. There should be no dispute that such push-based functionality was in the background knowledge of a POSA, given that Dr. Shenoy himself relies upon evidence that confirms this understanding. *See*, IETF HTTP/1.1, §§ 9.5, 9.6.

71. Dr Shenoy is incorrect in attempting to draw a bright line between embodiments of Sigurdsson that use a peer-to-peer architecture and embodiments that use a client-server (“drop box”) architecture. *See* Shenoy Decl., ¶ 80. As discussed above, Sigurdsson discloses embodiments that intermingled and describe as interchangeable the use of peer-to-peer concepts and client-server concepts. For example, paragraph 79 of Sigurdsson discloses the use of peer-to-peer file transfers and states: “Additionally, the Client C 406 can request the content from the Server *even if the one or more other clients are online*. For instance, the Client C may request the content from the Server if the other clients have less available bandwidth relative to the Server.” Thus, Sigurdsson itself provides motivation in the form of

“less available bandwidth” for using Server 106 as a “drop box” in a client-server paradigm *even when other clients are online and available to transfer a file to a client that needs it.*

72. As such, Dr. Shenoy is simply wrong in stating that “the peer to peer architecture and client server architecture are two fundamentally different methods for file access and are typically used in different situations.” Shenoy Decl., ¶ 82. To the contrary, paragraph 79 of Sigurdsson makes it clear that they are interchangeable and may be used in the same situations. Thus, it would have been obvious to use the peer-to-peer concepts disclosed in Sigurdsson with embodiments that are primarily focused on a client-server implementation and vice versa.

73. For the foregoing reasons and those in my Opening Declaration, this limitation would have been obvious in view of Sigurdsson alone. However, this limitation would have also been obvious in view of Sigurdsson in combination with Shappell. Balakrishnan Open. Decl., ¶¶ 531-33. Shappell expressly teaches and would have motivated a POSA to implement this limitation in view of Shappell’s “automatic replication” feature.

74. As discussed in my Opening Declaration, Shappell teaches a user-selectable “File Replication” setting that may be used to switch between (1) manually downloading files “on an on-demand basis,” and (2) automatically downloading files when available, with an “automatic replication” setting. Shappell,

¶¶ 62, 81. When “automatic replication” is enabled, Shappell teaches automatically replacing an “out-of-date file” with an “updated version” on a receiving computer. *Id.*, ¶ 81 (“When a modification of a file contained in a shared space occurs, a notification to other members in the group occurs (see steps 1902, 1904 and 1906 in FIG. 19). In a preferred embodiment, upon receipt of such notification, member machines that have previously obtained a copy of the shared file will ***remove the local copy of the out-of-date file. . . . [I]f automatic replication of the shared file is enabled, the local machine obtains an updated version of the file***, as is shown in step 2006 in FIG. 20.”), Figs. 19-20.

75. It would have been obvious to implement Shappell’s “automatic replication” feature as an option for each Sigurdsson client device, including Client C (the “third electronic device”), so that Server 106 would ***automatically*** transfer to Client C the copy of a file modified at Client A (the “second electronic device”) to replace Client A’s older version of the file, to ensure that the user at Client C would have the most recently updated file. A POSA would have been motivated to incorporate Shappell’s auto-replication to address Sigurdsson’s concern for each client “to keep its locally stored content synchronized with content from the other clients,” and it easily and obviously would be implemented applying Sigurdsson’s teaching that each client can “maintain an open connection” with the server. Sigurdsson, ¶¶ 84-85.

76. Dr. Shenoy is incorrect in asserting that “it would not have been obvious to a POSA to combine Sigurdsson and Shappell because Shappell teaches away from using a server.” Shenoy Decl., ¶ 83. To the contrary, paragraph 79 of Sigurdsson teaches that use of a server is interchangeable with use of a client in a peer-to-peer setup. Sigurdsson, ¶ 79 (“Additionally, the Client C 406 can request the content from the Server *even if the one or more other clients are online*. For instance, the Client C may request the content from the Server if the other clients have less available bandwidth relative to the Server.”). Because Sigurdsson also discloses certain “server-less” embodiments involving direct file transfers from client to client, the POSA would have been motivated to look to other “server-less” embodiments such as those in Shappell, for the reasons discussed in my Opening Declaration. See Balakrishnan Open. Decl., ¶¶ 181-94.

77. Contrary to Dr. Shenoy’s misrepresentation of Shappell, there is nothing in Shappell that “discounts the use of a server[.]” Shenoy Decl., ¶ 85. In fact, Shappell does not specifically disparage a client/server implementation, but instead proclaims that the “type of network . . . is not critical” and is only “preferably” peer-to-peer, and suggests transferring “indirectly” between user machines such as would be provided by a client/server architecture (Shappell, ¶ 40). Moreover, there is nothing in the Challenged Claims that requires the use of a server, as discussed above—any electronic device may be used. In fact, Shappell expressly

discloses that the “remote computer 49” shown in Fig. 1 of Shappell, with which user “device 20” may communicate, may be “a server” *or*, in the alternative, a “peer device[.]” Shappell, ¶ 37. Thus, Shappell contemplates the same paradigm as Sigurdsson in which a server is treated the same as a peer client.

78. Dr. Shenoy’s statement that “it would not have been obvious to a POSA to have modified the teachings of Sigurdsson to provide a server-less shared space (i.e., replace the client/server mechanism) as disclosed in Shappell, because doing so would change the principle of operation of Sigurdsson[.]” is irrelevant because that is not what I or the Petitioners have suggested doing. Shenoy Decl., ¶ 88. As noted above and in my Opening Declaration, it would have been obvious in view of Shappell’s express disclosure of using a server or peer client as equally available and alternative options and Shappell’s “automatic replication” feature to automatically transfer a file updated on one client and stored at Server 106 to another client. Shappell’s “server-less” embodiments do not teach away from Sigurdsson’s embodiments that use a server any more than Sigurdsson’s own embodiments that do not include a server—a POSA would not have viewed peer-to-peer and client-server mechanisms to be incompatible with one another or to teach away from one another in view of Sigurdsson’s express teachings that they are interchangeable.

79. Specifically, a POSA would have understood the benefits of using a server as a “drop box” when a client is offline, as taught in Sigurdsson, and also

using Shappell’s automatic replication teaching (and Sigurdsson’s related teachings) to automatically transfer an updated file to the client when it comes back online. Nothing in Shappell would have made a POSA think that incorporating such functionality from Shappell in Sigurdsson would have “change[d] the principle of operation of Sigurdsson[,]” as Dr. Shenoy erroneously suggests. Shenoy Decl., ¶ 89.

**b. Sigurdsson Teaches and Shappell Does Not Teach
Away from the Second and Third Electronic Devices
Being Associated with the Same User**

80. Claim 1 of the ’561 patent recites that the “second” and “third” electronic devices are associated with a user. Dr. Shenoy does not dispute that Sigurdsson expressly discloses these concepts, as discussed in detail in my Opening Declaration. *See, e.g.*, Balakrishnan Open. Decl., ¶¶ 120-21, 159. Instead, he argues that: “A POSA would not find that Shappell’s sharing files **among group members** in the shared space teaches the **associated with the same user** limitation. While Shappell discloses sharing a file with members of a group, Shappell simply does not send the file to a specific device ‘associated with the user.’” Shenoy Decl., ¶ 93 (emphasis in original). Dr. Shenoy does not argue that Shappell teaches away from Sigurdsson’s express disclosures regarding these concepts. A POSA would not have been dissuaded by anything in Shappell from synchronizing files of the same user across multiple user devices as taught in Sigurdsson, particularly in view of Sigurdsson’s explanation of the obvious benefits of keeping a user’s files

synchronized across multiple user devices, such as a user's home and work computers. *See, e.g.*, Sigurdsson, ¶ 26.

81. Moreover, I disagree with Dr. Shenoy's characterization of Shappell. The fact that group members in Shappell's shared spaces may share files between each other necessarily means that the computers involved in such synchronization are associated with the user in the sense that the user is part of the same group in which the files are shared and synchronized. In fact, Shappell expressly discloses predetermination of "who can be members" of the shared group using a group policy associated with the user that would be applied to "all the machines in the domain to which the policy pertains." Shappell, ¶ 87. Thus, all machines in the particular domain would be associated with the user in the sense that they adopt the same group policy as the user's device.

C. Claims 1, 3, 4, 10, 12 and 13 of the '942 Patent Would Have Been Obvious in View of Sigurdsson in Combination with Shappell.

1. Independent Claim 1

- a. "automatically send[ing], via the first application, the modified first electronic file copy to a third application at the third electronic device . . . responsive to receiving the modified first electronic file copy from the second electronic device"**

82. Dr. Shenoy opines that Sigurdsson does not disclose that "automatically send[ing], via the first application, the modified first electronic file copy to a third application at the third electronic device . . . responsive to receiving the modified

first electronic file copy from the second electronic device[,]” as this limitation appears in claim 1 of the ’942 patent (similar limitations appear in each of the Challenged Claims). Shenoy Decl., ¶ 136. I disagree. In my opinion, Sigurdsson expressly discloses this claim element. *See, e.g.,* Sigurdsson, ¶ 26 (“Exchange of content between the computer devices can occur automatically without a user specifying that the content should be sent to another computer device.”). In addition, even if Sigurdsson does not disclose this claim element, it at least suggests or renders the element obvious, as discussed below. In addition, this element is rendered obvious in view of Sigurdsson in combination with Shappell, for the reasons discussed below and in my Opening Declaration.

83. Dr. Shenoy argues that this claim element requires the server to automatically transfer a modified file without a request from the client for the modified file. Shenoy Decl., ¶ 137. I disagree. Nothing in the claims or specification requires transfer of the modified file from the server to the client without a request from the client for the modified file. In fact, the Challenged Patents reference “push” and “pull” approaches as if they are trivially interchangeable, and a POSA would understand from reading such disclosure that the claims do not require a server to push the modified first electronic file copy from the first electronic device to the third electronic device:

While a preferred embodiment of the invention has been illustrated and

described, as noted above, many changes can be made without departing from the spirit and scope of the invention. For example, as an alternative to the approach described with reference to FIG. 3, wherein the transfer clients 310, 320 function to “push” modified or created files to the synch application 380, the synch application 380 may instead function to periodically “pull” or otherwise actively retrieve such files from the transfer clients 310, 320[.] Instead, the invention should be determined entirely by reference to the claims that follow.

'942 patent, 10:48-58. The '942 and '561 patents' inventors knew how to distinguish between “push” and “pull” data transfer models based on the above disclosures but nevertheless decided not to limit the claims of the patents to require use of either one of these models by not using the word “push” (or the concept itself) in any of the Challenged Claims. Thus, Dr. Shenoy is wrong to simply assume that a push model is required, particularly given that he cites no language from the Challenged Patents requiring such an interpretation.

84. One part of the '942 and '561 patents' specification states:

Typically, the server 230 will store one or more of the received files to the storage 270 only if the destination user device is offline or otherwise temporarily not in communication with the server 230. Upon resuming communication with the destination user device, the server 230 will transfer the temporarily stored files to the destination device.

'942 patent, 8:26-31. In my opinion, a POSA would understand this disclosure to

contemplate a previously-offline user device first requesting to resume communication with the server before the server sends a modified file to the user device. Thus, Dr. Shenoy is wrong in assuming the claims require the absence of a request from a user device before a server sends a modified file to the user device.

85. The Challenged Patents expressly contemplate user devices connecting to a server via the internet. '942 patent, 7:28-32. As such, a POSA would have understood the Challenged Claims to cover the situation in which a previously-offline user device reconnects to the server via a public network over the internet using DHCP. For the reasons discussed above in § VI, the user device would have been **required** in this scenario to first reestablish communication with the server (not the other way around). Thus, Dr. Shenoy is incorrect in interpreting the Challenged Claims to require the **server** to transfer a modified file to a “third electronic device” without a request from the client. Shenoy Decl., ¶ 137.

86. Dr. Shenoy incorrectly states that “Sigurdsson describes that its system is implemented as an HTTP client/server operating based on request/response process.” *Id.* To the contrary, Sigurdsson expressly describes that the client user devices described in Sigurdsson **and** the server may **each individually** include an HTTPS client **and** an HTTPS server. This is shown, for example, in Fig. 3 of Sigurdsson, which depicts “Client B” as including both an HTTPS client and server that bidirectionally communicate with a “Server” that a POSA would understand to

include a reciprocal HTTPS server and client. *See also* Sigurdsson, Fig. 3, ¶ 47 (“The HTTPS Client/server 326 **can be used as a client** that tracks web access and can contain a Time Stamp 328 which associates viewed web pages with a time that they were accessed by the user. The HTTPS Client/server **can also be used as a server** to provide content to peer clients, which is described in more detail below.”). Sigurdsson further explains that all of its clients are fungible (i.e., they contain the same components), and Dr. Shenoy does not dispute this point. *E.g.*, Balakrishnan Open. Decl., ¶ 122. Thus, a POSA reading Sigurdsson would understand that clients can contain both HTTPS clients and servers that must necessarily communicate with a reciprocal HTTPS client or server on, e.g., Server 106 depicted in Fig. 3 of Sigurdsson.

87. Furthermore, Sigurdsson explains that “the HTTPS Client/server” “can transmit an HTTP post command containing the Client X Content 410 to the Server 106.” Sigurdsson, ¶ 54. Because the server may communicate with each client bidirectionally, and because each client is expressly described in Sigurdsson as including both an HTTPS client **and** an HTTPS server, a POSA would have understood or at least found it obvious from these disclosures that the client HTTPS servers could be used by, e.g., “Server 106” to transfer a modified file to a client device.

88. Although Sigurdsson describes use of clients’ HTTPS servers in certain

embodiments involving peer-to-peer communication between clients, Sigurdsson also states that, even in such peer-to-peer implementations, “[t]he Server 106 can still serve as a drop-box for temporarily (or permanently) storing Client X Content 410 originating from any [of] the Clients 102, 104 or 406.” *Id.*, ¶ 51. Thus, a POSA would understand that Server 106 could communicate with clients in the same manner that clients communicate with each other, namely, that the Server could send a modified file to a client.

89. In fact, this is precisely what Sigurdsson contemplates in the alternative embodiment of step 606: “the Client X Content 410 can be sent directly to the Server 106, thus bypassing the Client B 104. ***The Server 106 can then transmit the content to the remaining client peers, such as the Clients B and C.***” Sigurdsson, ¶ 68. As noted by the Board in its institution decision for the ’942 patent, “Sigurdsson . . . does not describe the transfer of a modified file with the alternative embodiment of step 606 as resulting from any request from Client C.” IPR2023-00433, Paper 17 (’942 Patent Institution Decision) at 36-37; IPR2023-00427, Paper 17 (’561 Patent Institution Decision) at 34.

90. Dr. Shenoy is incorrect in stating that HTTP protocol did not support push-based functionality at the time of the invention. Shenoy Decl., ¶ 137. To the contrary, as discussed above in § VI, given that Sigurdsson expressly describes its clients as including HTTPS servers (in addition to clients), a POSA would have

understood and found it obvious to use Server 106 to push a modified file to the client.

91. The Challenged Claims and specification do not require the use of a “push” model of data transfer or the absence of intervening steps between when a modified file is received from one user device at Server 106 and when it is forwarded by Server 106 to another user device. For example, claim 4 of the ’561 patent expressly requires the “first application” (on, e.g., Server 106 in Sigurdsson) to “store the modified first electronic file copy to the memory device associated with the first electronic device when the third electronic device is not in communication with the first electronic device.” The Challenged Patents describe a “synch application” that “polls” client devices to determine whether they are in communication with the server running the synch application. ’942 patent, 8:50-54. Thus, claim 4 and the Challenged Patents’ specification contemplates at least one intervening step that does not appear in claim 1 but is required before a file may be transferred to the third electronic device in claim 4—namely, determining whether the first electronic device is in communication with the third electronic device. Given that claim 4 depends (indirectly) from claim 1, it necessarily follows that claim 1 must include within its scope implementations that do not immediately forward modified files from one client device through a server to another client device but, instead, implement other steps in between, such as allowing a previously-

offline client to reestablish communication with the server, request a list of missing content, and then request the missing content itself.

92. Dr. Shenoy opines: “A POSA, considering the [Sigurdsson] reference as a whole, would not have understood the discussion of the alternate embodiment of step 606 described in ¶68, and the alternate embodiment of step 620 described in ¶80 to be divorced from the accompanying steps in Fig. 6.” Shenoy Decl., ¶ 139. I understand from counsel that a claim is obvious if a POSA would have been motivated to combine teachings from various embodiments in a single prior art reference based on the teachings of those embodiments, the general knowledge present in the art, and common sense. I have also been informed by counsel that a POSA has ordinary creativity, and that there is no requirement that there be an actual or explicit teaching to combine two embodiments in a single prior art reference. Dr. Shenoy does not appear to be applying these principles from the perspective of a POSA; instead, he is analyzing only the precise embodiment associated with Fig. 6 in Sigurdsson in isolation, which I understand is not the correct analysis.

93. A POSA would have read the steps associated with Fig. 6 in Sigurdsson in view of other disclosures in Sigurdsson. For example, Sigurdsson discloses:

The Server 106 can facilitate synchronization of content among the user's computers. When the Client A 102 (e.g., the user's home computer) connects to the Server 106, Client B Content 112 originating

from the Client B 104 is automatically copied to the Client A 102, as indicated by arrow 116. In some implementations, copying occurs for the Client B Content 112 not already stored on the Client A 102. Similarly, whenever the Client B 104 (e.g., the user's work computer) connects to the Server 106, any missing Client A Content 108 is copied automatically to the Client B 104, as indicated by arrow 118.

Sigurdsson, ¶ 31. A POSA reading this disclosure would have found it obvious to automatically transfer a modified file from Client A to the Server 106 and then from Server 106 to Client C, upon detecting that Client C comes back online when reading the embodiment associated with Fig. 6, as the concept of automatically transferring files for purposes of synchronization across multiple user devices as soon as possible is clearly reflected in the above excerpt (as well as many others in Sigurdsson). I disagree with Dr. Shenoy's approach of analyzing the embodiment associated with Fig. 6 of Sigurdsson without considering other disclosures in Sigurdsson. In analyzing paragraphs 68 and 80 in Sigurdsson, I have at all times considered the teachings of Sigurdsson as a whole—not solely the embodiment associated with Figure 6.

94. In paragraph 141 of his declaration, Dr. Shenoy implies that paragraphs 61-82 of Sigurdsson are part of a single sequence. Shenoy Decl., ¶ 141. I disagree—the paragraphs instead describe a set of alternative ideas that may be used interchangeably in a “mixed peer-to-peer and server architecture[.]” Sigurdsson,

¶ 61; *see also id.*, ¶ 62 (“The system 400 can enable peer-to-peer content sharing and synchronization using a combination of peer-to-peer and ‘drop box’ techniques.”). A POSA would have been motivated to use the ideas explained in connection with Fig. 6 of Sigurdsson interchangeably with any of the embodiments described in Sigurdsson when, for example, such interchangeable use would result in obvious benefits to users of the systems and methods described therein, as more fully discussed in my Opening Declaration.

95. Paragraph 61 of Sigurdsson explains that, like the inventions described in the ’942 and ’561 patents’ embodiments, “polling” may be used to determine whether a client is offline or online, which confirms that a POSA would not read the Challenged Claims to require the absence of any user request before an updated file is automatically sent to a client from a server. *See* Sigurdsson, ¶ 61 (“In one implementation, determination of whether a client is online or offline can be made by monitoring polling by the client. If no *request* is received from a client, then the client is assumed to be offline. Similarly, the Server 106 can maintain a table of client IDs and whether each client is online or offline. When a client comes online, it can notify the Server 106 that it is online, and the Server 106 can update its table of clients with the online status.”).

96. In connection with the embodiment associated with Fig. 6, Sigurdsson first states that “a user at the Client A 102 may edit and save a document on his work

computer.” Sigurdsson, ¶ 63. Thereafter, Fig. 6 includes a “step 606” that may be implemented in two different alternative ways: (1) Client A may directly send its content to another client (e.g., Client B), as described in paragraph 65 of Sigurdsson, or (2) Client A may send its content “directly to the server,” as described in paragraph 68 of Sigurdsson. Thereafter, “the Client X Content 410 can be sent directly to the Server 106, thus bypassing the Client B 104. ***The Server 106 can then transmit the content to the remaining client peers, such as the Clients B and C.***” Sigurdsson, ¶ 68.

97. Dr. Shenoy states: “While ¶68 states that the ‘Sever [sic] 106 can then transmit the content to the remaining client peers, such as Clients B and C’ it in no way teaches or suggests that Server 106 *automatically* transfers the content to a client.” Shenoy Decl., ¶ 143 (emphasis in original). I disagree. In my opinion, paragraph 68 of Sigurdsson ***expressly*** teaches automatically propagating content received from Client A at Server 106 to another client, even under Dr. Shenoy’s flawed understanding of what is required by the Challenged Claims. The paragraphs that follow paragraph 68 do not relate to the embodiment described in paragraph 68, in which content received from Client A is sent by Server 106 to Clients B and C, given that Client C is offline in the paragraphs that follow paragraph 68. Therefore, I disagree with Dr. Shenoy’s statement that “the very next paragraph [after paragraph 68] begins the discussion of *how* Server 106 transfers the content, through steps 612

to 626, when Client C comes back online.” *Id.* (emphasis in original).

98. I also disagree with Dr. Shenoy’s statement that “[a] POSA would understand that Sigurdsson at ¶ 68 discloses that the alternative embodiment of step 606 is part of the overall sequence for handling synchronization *when a client is offline* when a transfer of content is to take place.” Shenoy Decl., ¶ 144 (emphasis in original). To the contrary, paragraph 68 expressly discussed the Server 106 sending content to Client C, which would, of course, be impossible if Client C were offline (as it is described in the paragraphs that follow paragraph 68).

99. I likewise disagree with Dr. Shenoy’s statement that, “[c]onsidering the Sigurdsson reference as a whole, a POSA would have considered Sigurdsson’s alternative embodiment of step 606 as limiting the transfer of the content to using step 620 and its associated steps 612-620 and 626-630.” *Id.*, ¶ 145. To the contrary, Fig. 6 includes a horizontal, dashed line that clearly separates the alternative embodiment of step 606 described in paragraph 68 from the remaining steps shown in Fig. 6. This is because Client C is online, not offline, in paragraph 68’s embodiment in which “the Client X Content 410 can be sent directly to the Server 106” and thereafter “[t]he Server 106 can then transmit the content to the remaining client peers, such as the Clients B and C.” Sigurdsson, ¶ 68. Dr. Shenoy ignores this text in paragraph 68, pretending that, instead, paragraph 68 has something to do with the steps that follow paragraph 68 when Client C is offline. This is incorrect.

100. The alternative embodiment for step 606 described in paragraph 68 also contradicts Dr. Shenoy's incorrect statement that, "[i]f Client C was online, Sigurdsson teaches that the transfer would be performed using the peer-to-peer technique that does not involve sending content from the server." Shenoy Decl., ¶ 145. This is clearly incorrect—paragraph 68 expressly contemplates Client C being **online** (otherwise, it could not receive content from Server 106) and expressly contemplates Client A content stored at Server 106 being sent to Clients B and C. Paragraph 68 also contradicts Dr. Shenoy's flawed assumption that all transfers to a client "would not be performed by the server using the 'drop box' technique" when a sending client and receiving client are online. Shenoy Decl., ¶ 145. To the contrary, paragraph 68 makes it clear that the peer-to-peer and drop-box techniques may be employed interchangeably when sending and receiving clients are online—the sending client may either send to Server 106, which can then forward the file to the receiving client, or the sending client may send to the receiving client directly. A POSA would have viewed either of these options as obvious, conventional, and well-known design choices that would have been obvious to try in view of Sigurdsson's disclosures.

101. The alternative embodiment for step 606 described in paragraph 68 also contradicts Dr. Shenoy's incorrect statement that, "[w]hile it is true that Sigurdsson discloses that the Server 106 can transmit the content to the remaining client peers,

such as the Clients B and C, Sigurdsson is clear that this transmission of the content by the Server 106 is performed *only* after receiving a request for the content from the Clients B and C.” Shenoy Decl., ¶ 147 (emphasis in original). There is no such “request for the content from the Clients B and C” disclosed in paragraph 68, and Dr. Shenoy is wrong to read paragraph 68 as requiring the request from Client C discussed in the paragraphs that follow paragraph 68 in Sigurdsson.

102. I disagree with Dr. Shenoy’s statement that “[a] person of ordinary skill in the art would not understand the statement in ¶80 [in Sigurdsson] to teach or suggest Server 106 automatically sending the modified content to Client C.” Shenoy Decl., ¶ 146. Paragraph 80 of Sigurdsson discloses: “In step 626, the server sends the missing content to the third client. For example, referring to FIG. 4, the Server 106 sends the Client X Content 410 to the Client C 406.” Sigurdsson, ¶ 80. In Fig. 6, step 626 is depicted as occurring after Client C comes back online (after being offline) and after Client C requests and receives a list of missing content and requests missing content from the server in step 626. Sigurdsson, ¶¶ 69-80. Again, I read the disclosures in paragraphs 69-80 to be a recitation of interchangeable ideas and concepts in view of which the POSA would have found it obvious to automatically send content from Client A at Server 106 to Client C after Client C comes back online.

103. In addition, for the reasons discussed above, a POSA would understand

that Server 106 automatically sends the Client A content to Client C, even though Client C requests a list of missing content and requests the missing Client A content from Server 106. A POSA would view the sending of Client A content from the server to Client C as described in paragraphs 69-80 to be automatic both from the server's perspective and the user's perspective. Even if Sigurdsson's systems and methods are only automated from the user's perspective, that is all that the Challenged Claims require in view of Dr. Shenoy's opinion that the Challenged Claims are "user-centric." Shenoy Decl., ¶ 22. The intervening steps do not change the automatic nature of sending the Client A content by the server to Client C. As noted above, all independent claims recite "comprising," which I understand to permit additional steps such as those involving Client C requesting a list of missing content and requesting the missing content from the server while still practicing the Challenged Claims.

104. In paragraph 53 of his declaration, Dr. Shenoy presents a false dichotomy between "a first scenario when **clients are currently online** and a second scenario when the offline **client comes online again**" in which, in the first scenario (according to Dr. Shenoy), a peer-to-peer model must be used and in which, in the second scenario (according to Dr. Shenoy), the "drop box" model must be used. Shenoy Decl., ¶ 150 (emphasis in original). There is no disclosure in Sigurdsson that, when clients are online, they *must* use a peer-to-peer paradigm, whereas, when

clients are offline, they ***must*** use the “drop box” paradigm. In fact, paragraph 68 of Sigurdsson is an express example to the contrary—it expressly discloses sending content received from Client A at Server 106 to Clients B and C (which obviously must be online to receive the content). Sigurdsson, ¶ 68. In paragraph 150, Dr. Shenoy again misrepresents Sigurdsson’s disclosure in paragraph 68 as requiring a request from a client before the server may send content to the client (Shenoy Decl., ¶ 150)—but there is no such disclosure in paragraph 68, and no other paragraph of Sigurdsson so limits paragraph 68.

105. Dr. Shenoy incorrectly asserts that “the first scenario in which the client is already online and Sigurdsson at ¶ 65 unambiguously states that the peer-to-peer technique should apply.” Shenoy Decl., ¶ 152. This sentence effectively ignores paragraph 68, which is expressly described in Sigurdsson as an “alternative embodiment of step 606” described in paragraph 65. The peer-to-peer embodiment in paragraph 65 is only one alternative of multiple, with paragraph 68 disclosing an alternative embodiment that uses Server 106. Paragraph 68 both expressly discloses and suggests to a POSA that the Server 106 automatically transfers content 410 to Client C 406 without receiving a request from Client C 406—no such request is referenced in paragraph 68 of Sigurdsson.

106. Dr. Shenoy again mischaracterizes Sigurdsson in his statement: “Therefore, in every case involving a Server 106 transmitting content to the Client

C, Sigurdsson requires the Server 106 to receive a request from the Client C before the Server 106 transmits the content 106 to the Client C.” Shenoy Decl., ¶ 157. Again, the alternative embodiment of step 606 in paragraph 68 requires no “request from the Client C before the Server 106 transmits the content 106 to the Client C”—such a requirement simply does not appear in paragraph 68. In addition, as discussed below, even if a request were required from a client before the server sends content to the client, it is my opinion that Sigurdsson still discloses the limitation requiring automatic transfer of a modified file from the “first electronic device” to a “third electronic device” (as recited in claim 1 of the ’561 patent). For the foregoing reasons, and contrary to Dr. Shenoy’s assertions, a POSA would *not* have “consider[ed] Sigurdsson’s disclosure to limit the transfer described for the alternative embodiment of step 606 to step 620 and its associated steps 612-618 and 626-630.” Shenoy Decl., ¶ 158.

107. Dr. Shenoy opines that, “even if Sigurdsson were read to have one client send the modified content to the server to synchronize with another client that was online at the time, this hypothetical example would fall outside the scope of claims 4 and 11.” Shenoy Decl., ¶ 158. As I explained in my Opening Declaration (¶¶ 543, 550), claims 4 and 11 of the ’561 patent would have been obvious in view of the combination of Sigurdsson’s disclosures in paragraphs 66 and 68. In particular, paragraph 68 states that Client A may send data “directly to the server”

and paragraph 66 states: “in this case, *the third client is offline, so the content is transmitted to the server for temporary storage.*” Sigurdsson, ¶¶ 66, 68. Thus, Sigurdsson, clearly suggests and renders obvious the concept of temporarily storing content at the server when Client C is offline.

108. In addition, as I explained in my Opening Declaration, the limitation for using the “first application” on the first electronic device to “automatically transfer the modified first electronic file copy to the third electronic device” would have been obvious in view of other Sigurdsson disclosures. ’561, claim 1. Sigurdsson’s Server 106 in Figure 4 includes a “Synchronization Manager” that “synchronize[s] content stored on and shared between” different clients (e.g., “Client A” and “Client C”). Sigurdsson, ¶ 48. As noted in my Opening Declaration, a POSA would understand that “synchronization” in this context means “version comparisons of copies of . . . files to ensure they contain the same data.” Balakrishnan Open. Decl., ¶ 522 (quoting EX1021, Microsoft Computer Dictionary (5th ed. 2002), 506 (entry for “synchronization”).

109. The steps associated with Figure 6 must be read in the context of these disclosures related to Fig. 4. Sigurdsson, ¶ 61 (“**FIG. 6** is another sequence diagram showing an illustrative method for sharing and synchronizing content using a mixed peer-to-peer and server architecture of **FIG. 4** when a client that is offline comes online.”). In particular, the steps in Fig. 6 describe an automated process for

(1) detecting when Client C comes back online, (2) passing a list containing metadata to enable Client C to see what content it is missing, (3) receiving a request from Client C for missing data, and (4) sending the missing data to Client C. Balakrishnan Open. Decl., ¶ 523. A POSA would have understood the steps in paragraphs 69-81 of Sigurdsson to be an automated process that includes sending Client A content stored at Server 106 to Client C when Client C comes back online. *See, e.g.*, Sigurdsson, ¶ 79 (“Additionally, the Client C 406 can request the content from the Server even if the one or more other clients are online. For instance, the Client C may request the content from the Server if the other clients have less available bandwidth relative to the Server.”).

110. As discussed above, the fact that Client C may send a request for the Client A content stored at the server before receiving it in paragraphs 69-81 does not change the automated nature of the process described in these paragraphs. None of the steps in paragraphs 69-81 are described as being performed by the user or requiring user input (or input from anything else). For example, paragraph 69 states that Client C “requests a list of missing content from the Server 106 *when it connects to network that links the server and other clients, such as the Internet.*” Sigurdsson, ¶ 69. A POSA would understand that this is how a typical client using DHCP would connect to the server (via a request from the client to the server) for the reasons discussed above—e.g., the server would not know the client’s IP address

when the client connects to a public network on the internet. Thus, paragraph 69 makes it clear that Client C’s request for a list of missing content is contemporaneous with Client C’s connection to the server—a POSA would understand that this functionality would necessarily be automated from both the user’s and the server’s perspectives.

111. Paragraphs 70-71 describe an automated process for determining the list of content that may be missing from Client C or that Client C may request using an automated “priority formula.” Sigurdsson, ¶¶ 70-71. Again, the POSA would understand these steps to be automated from both the server and user’s perspective—the client automatically determines what to request without the need for any outside input. Paragraphs 73-74 note that the list of missing content may specify content that Client C “can obtain from the server.” *Id.*, ¶¶ 73-74. In one alternative embodiment of step 620, Client C “can obtain the missing content from the server.” *Id.*, ¶ 79. In paragraphs 80-81, the Client A content stored on Server 106 is sent to Client C from the server and stored at Client C. *Id.*, ¶¶ 65, 68, 80-81. A POSA would understand that all of these steps would be implemented in software and automated from both the server’s and user’s perspectives.

112. In connection with Fig. 1, Sigurdsson discloses:

The Server 106 can facilitate synchronization of content among the user’s computers. When the Client A 102 (e.g., the user’s home

computer) connects to the Server 106, Client B Content 112 originating from the Client B 104 is ***automatically*** copied to the Client A 102, as indicated by arrow 116. In some implementations, copying occurs for the Client B Content 112 not already stored on the Client A 102. Similarly, whenever the Client B 104 (e.g., the user's work computer) connects to the Server 106, any missing Client A Content 108 is copied ***automatically*** to the Client B 104, as indicated by arrow 118.

Sigurdsson, ¶ 31. Sigurdsson later explains that Fig. 2 is a “sequence diagram showing an illustrative method 200 for sharing and synchronizing content across multiple client devices ***according to the implementation shown in FIG. 1.***” *Id.*,

¶ 32. Thus, Fig. 2's embodiment is automated like the embodiment in Fig. 1. Likewise, Sigurdsson explains that “FIG. 4 is a schematic showing a system 400 for sharing and synchronizing content” and that, in its depicted “system 400,” “content still can be shared among the Client A 102, the Client B 104 and a Client C 406 using the Server 106 ***in a method similar to that of method 200 of FIG. 2.***” *Id.*, ¶

51. Thus, Fig. 4's synchronization embodiments are also automated like those associated with Figs. 1 and 2. Finally, Sigurdsson states: “FIG. 6 is another sequence diagram showing an illustrative method for sharing and synchronizing content ***using a mixed peer-to-peer and server architecture of FIG. 4*** when a client that is offline comes online.” Sigurdsson, ¶ 61. Because the synchronization associated with Fig. 1 is described as being automated, and because the methods of synchronization

associated with Figs. 2, 4, and 6 are described as operating similar to the synchronization associated with Fig. 1, a POSA would have understood Fig. 6's embodiment (described in paragraphs 61-81) to be automated.

113. For the reasons discussed above, I disagree with Dr. Shenoy's opinion that "Sigurdsson is directed to a pull mechanism, and fails to automatically send the modified first electronic file copy to the third electronic device associated with the user." Shenoy Decl., ¶ 160. Sigurdsson discloses that its clients and Server 106 includes **both** HTTPS clients **and** HTTPS servers for bidirectional communication, and paragraph 68 expressly discloses pushing content from Server 106 to Clients B and C. In addition, a POSA would have understood how to implement push-type data transfers including by using disclosures such as those in the Hepworth reference, discussed above. No modification of Sigurdsson would be necessary to implement push functionality—it is already expressly recited in paragraph 68. Server 106 is not described as an "HTTP server[,]” as Dr. Shenoy misleadingly suggests (*id.*, ¶ 162). Instead, as discussed above, a POSA would have understood Server 106 as including both an HTTPS client and server for bidirectional communication with clients' HTTPS clients and servers, including communications to clients from Server 106 to effectively push data/files to clients.

114. Dr. Shenoy's own cited evidence confirms that a POSA would have understood Server 106 to implement push-based data/file transfers to clients based

on Sigurdsson's disclosure of its clients including HTTPS servers, because a POSA would know that HTTPS servers may receive PUT and POST messages to effectively push data to an HTTPS server. IETF HTTP/1.1, § 9.5 ("The POST method is used to request that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line."); *id.*, § 9.6 ("The PUT method requests that the enclosed entity be stored under the supplied Request-URI.").

115. Moreover, the Challenged Claims do not recite the word "push" and do not require a push mechanism to transfer data from the server to any client. The inventors knew how to use the words "push" and "pull" to describe data transfers and purposely did not limit the Challenged Claims to being implemented via push or pull.

116. In addition, neither Sigurdsson nor Shappell *requires* use of HTTP/HTTPS, and a POSA would have known how to implement push-based file synchronization using push-based applications. The Challenged Patents refer to "a browser *or other applications*, running on one or more of the user devices 210, 280 [that] can cooperate in two-way communication with server 230 and one or more applications running on server 230." '942 patent, 7:51-54. The Challenged Patents provide no detail about how to implement any of these applications, confirming that doing so would have been routine, obvious, well-known, and conventional prior to

November 2008. Indeed, there were many well-known, conventional examples of applications used to push data from a server to a user's device that were in public use prior to November 2008, as discussed above in § VI with respect to the Wong reference.

117. Dr. Shenoy incorrectly states: “Sigurdsson at ¶ 31 does not correspond to the ‘automatic send’ limitation (*i.e.*, without a client request).” Shenoy Decl., ¶ 163 (emphasis in original). I disagree. Paragraph 31 clearly and unambiguously discloses an embodiment in which “Server 106 can facilitate synchronization of content among the user's computers” by automatically sending updated or new content from one client through Server 106 to another client. Sigurdsson, ¶ 31. There is nothing in Sigurdsson that limits the embodiment in paragraph 31 to any “pull” approach associated with Fig. 2 as Dr. Shenoy suggests. Shenoy Decl., ¶ 163. Instead, Sigurdsson makes clear that Fig. 2 describes one possible implementation of an embodiment using the system shown in Fig. 1, and that the client-request mechanism from Client B to the server, as shown in Fig. 2, is only one possible implementation of the automated synchronization process described in Fig. 1. Sigurdsson, ¶ 32. The association of Fig. 2 with Fig. 1's system, as discussed in paragraphs 31-32 of Sigurdsson, confirms that, even though Fig. 2's flow diagram includes a client request to the server, the flow diagram is still automated consistent with paragraph 31 and its description of Fig. 1.

118. There is nothing in the Challenged Patents or their file histories, such as a clear disclaimer or disavowal of claim scope or lexicography, that could justify limiting the claims to cover embodiments in which no client requests are sent to Sigurdsson's server. In fact, the Challenged Patents teach the opposite: a client *must* respond to a server's polling request to allow the server to first determine that the client is in communication with the server. '942 patent, 8:45-59. Even though the Challenged Claims do not expressly recite using such a client request, the Challenged Claims use the word "comprising," which means that Sigurdsson's embodiments disclose and render obvious the Challenged Claims even though they may contemplate certain automated communications between clients and the server that are not expressly recited in the Challenged Claims.

119. Dr. Shenoy incorrectly states: "Since Sigurdsson *expressly* states that Fig. 4 implements a method similar to Fig. 2, the method of Fig. 4 also includes steps are performed by the Client B104 to receive content." Shenoy Decl., ¶ 171 (emphasis in original). To the contrary, Sigurdsson states: "In the system 400, content still *can be* shared among the Client A 102, the Client B 104 and a Client C 406 using the Server 106 in a method similar to that of method 200 of FIG. 2." Sigurdsson, ¶ 51. The words "can be" are not the same as "must be." Thus, the Client B request shown in Fig. 2 is not required to implement steps associated with synchronization methods that use Fig. 4's system.

120. For the reasons above, the embodiments in paragraphs 31 and 68 of Sigurdsson make it clear that client requests are not necessary to implement the automatic sending of a modified file from the first electronic device to the third electronic device recited in claim 1 of the '561 patent. Dr. Shenoy is incorrect in opining to the contrary. *See* Shenoy Decl., ¶ 172. Moreover, as discussed above, the Challenged Claims cover the use of both pull and push mechanisms, so Dr. Shenoy's arguments that erroneously attempt to limit Sigurdsson's embodiments to using pull mechanisms are ultimately irrelevant.

121. Even if the Board were to conclude that one or all of Sigurdsson's embodiments are limited to pull-based HTTP mechanisms as Dr. Shenoy incorrectly asserts (*see* Shenoy Decl., ¶ 176), it would have been obvious in view of the embodiments disclosed in paragraphs 31 and 68 to push data from Server 106 to a client (as discussed in paragraph 47 of Sigurdsson). Use of such functionality would have been in the background knowledge of the POSA, as demonstrated by the fact that, e.g., paragraphs 45 and 47 of Sigurdsson disclose the use of HTTP "post" messages without even describing what they are. *See* Sigurdsson, ¶¶ 45, 47. The embodiments in paragraphs 31 and 68 of Sigurdsson describe pushing file updates from Server 106 to clients without the need to go into any detail about HTTP or other well-known file transfer methods that could have been used to implement Sigurdsson's systems and methods that implement such functionality. There should

be no dispute that such push-based functionality was in the background knowledge of a POSA, given that Dr. Shenoy himself relies upon evidence that confirms this understanding. *See* IETF HTTP/1.1, §§ 9.5, 9.6.

122. Dr Shenoy is incorrect in attempting to draw a bright line between embodiments of Sigurdsson that use a peer-to-peer architecture and embodiments that use a client-server (“drop box”) architecture. *See* Shenoy Decl., ¶ 177. As discussed above, Sigurdsson discloses embodiments that intermingled and describe as interchangeable the use of peer-to-peer concepts and client-server concepts. For example, paragraph 79 of Sigurdsson discloses the use of peer-to-peer file transfers and states: “Additionally, the Client C 406 can request the content from the Server ***even if the one or more other clients are online***. For instance, the Client C may request the content from the Server if the other clients have less available bandwidth relative to the Server.” Sigurdsson, ¶ 79. Thus, Sigurdsson itself provides motivation in the form of “less available bandwidth” for using Server 106 as a “drop box” in a client-server paradigm ***even when other clients are online and available to transfer a file to a client that needs it***.

123. As such, Dr. Shenoy is simply wrong in stating that “the peer to peer architecture and client server architecture are two fundamentally different methods for file access and are typically used in different situations.” Shenoy Decl., ¶ 179. To the contrary, paragraph 79 of Sigurdsson makes it clear that they are

interchangeable and may be used in the same situations. Thus, it would have been obvious to use the peer-to-peer concepts disclosed in Sigurdsson with embodiments that are primarily focused on a client-server implementation and vice versa.

124. For the foregoing reasons and those in my Opening Declaration, this limitation would have been obvious in view of Sigurdsson alone. However, this limitation would have also been obvious in view of Sigurdsson in combination with Shappell. Balakrishnan Open. Decl., ¶¶ 531-33. Shappell expressly teaches and would have motivated a POSA to implement this limitation in view of Shappell’s “automatic replication” feature.

125. As discussed in my Opening Declaration, Shappell teaches a user-selectable “File Replication” setting that may be used to switch between (1) manually downloading files “on an on-demand basis,” and (2) automatically downloading files when available, with an “automatic replication” setting. Shappell, ¶¶ 62, 81. When “automatic replication” is enabled, Shappell teaches automatically replacing an “out-of-date file” with an “updated version” on a receiving computer. *Id.*, ¶ 81 (“When a modification of a file contained in a shared space occurs, a notification to other members in the group occurs (see steps 1902, 1904 and 1906 in FIG. 19). In a preferred embodiment, upon receipt of such notification, member machines that have previously obtained a copy of the shared file will ***remove the local copy of the out-of-date file. . . . [I]f automatic replication of the shared file***

is enabled, the local machine obtains an updated version of the file, as is shown in step 2006 in FIG. 20.”), Figs. 19-20.

126. It would have been obvious to implement Shappell’s “automatic replication” feature as an option for each Sigurdsson client device, including Client C (the “third electronic device”), so that Server 106 would ***automatically*** transfer to Client C the copy of a file modified at Client A (the “second electronic device”) to replace Client A’s older version of the file, to ensure that the user at Client C would have the most recently updated file. A POSA would have been motivated to incorporate Shappell’s auto-replication to address Sigurdsson’s concern for each client “to keep its locally stored content synchronized with content from the other clients,” and it easily and obviously would be implemented applying Sigurdsson’s teaching that each client can “maintain an open connection” with the server. Sigurdsson, ¶¶ 84-85.

127. Dr. Shenoy is incorrect in asserting that “it would not have been obvious to a POSA to combine Sigurdsson and Shappell because Shappell teaches away from using a server.” Shenoy Decl., ¶ 180. To the contrary, paragraph 79 of Sigurdsson teaches that use of a server is interchangeable with use of a client in a peer-to-peer setup. Sigurdsson, ¶ 79 (“Additionally, the Client C 406 can request the content from the Server ***even if the one or more other clients are online***. For instance, the Client C may request the content from the Server if the other clients

have less available bandwidth relative to the Server.”). Because Sigurdsson also discloses certain “server-less” embodiments involving direct file transfers from client to client, the POSA would have been motivated to look to other “server-less” embodiments such as those in Shappell, for the reasons discussed in my Opening Declaration. *See* Balakrishnan Open. Decl., ¶¶ 181-94.

128. Contrary to Dr. Shenoy’s misrepresentation of Shappell, there is nothing in Shappell that “discounts the use of a server[.]” Shenoy Decl., ¶ 182. In fact, Shappell does not specifically disparage a client/server implementation, but instead proclaims that the “type of network . . . is not critical” and is only “preferably” peer-to-peer, and suggests transferring “indirectly” between user machines such as would be provided by a client/server architecture (Shappell, ¶ 40). Moreover, there is nothing in the Challenged Claims that requires the use of a server, as discussed above—any electronic device may be used. In fact, Shappell expressly discloses that the “remote computer 49” shown in Fig. 1 of Shappell, with which user “device 20” may communicate, may be “a server” *or*, in the alternative, a “peer device[.]” Shappell, ¶ 37. Thus, Shappell contemplates the same paradigm as Sigurdsson in which a server is treated the same as a peer client.

129. Dr. Shenoy’s statement that “it would not have been obvious to a POSA to have modified the teachings of Sigurdsson to provide a server-less shared space (i.e., replace the client/server mechanism) as disclosed in Shappell, because doing

so would change the principle of operation of Sigurdsson[,]” is irrelevant because that is not what I or the Petitioners have suggested doing. Shenoy Decl., ¶ 185. As noted above and in my Opening Declaration, it would have been obvious in view of Shappell’s express disclosure of using a server or peer client as equally available and alternative options and Shappell’s “automatic replication” feature to automatically transfer a file updated on one client and stored at Server 106 to another client. Shappell’s “server-less” embodiments do not teach away from Sigurdsson’s embodiments that use a server any more than Sigurdsson’s own embodiments that do not include a server—a POSA would not have viewed peer-to-peer and client-server mechanisms to be incompatible with one another or to teach away from one another in view of Sigurdsson’s express teachings that they are interchangeable.

130. Specifically, a POSA would have understood the benefits of using a server as a “drop box” when a client is offline, as taught in Sigurdsson, and also using Shappell’s automatic replication teaching (and Sigurdsson’s related teachings) to automatically transfer an updated file to the client when it comes back online. Nothing in Shappell would have made a POSA think that incorporating such functionality from Shappell in Sigurdsson would have “change[d] the principle of operation of Sigurdsson[,]” as Dr. Shenoy erroneously suggests. Shenoy Decl., ¶ 186.

**b. Sigurdsson Teaches and Shappell Does Not Teach
Away from the Second and Third Electronic Devices
Being Associated with the Same User**

131. Claim 1 of the '561 patent recites that the “second” and “third” electronic devices are associated with a user. Dr. Shenoy does not dispute that Sigurdsson expressly discloses these concepts, as discussed in detail in my Opening Declaration. *See, e.g.*, Balakrishnan Open. Decl., ¶¶ 120-21, 159. Instead, he argues that: “A POSA would not find that Shappell’s sharing files **among group members** in the shared space teaches the **associated with the same user** limitation. While Shappell discloses sharing a file with members of a group, Shappell simply does not send the file to a specific device ‘associated with the user.’” Shenoy Decl., ¶ 190 (emphasis in original). Dr. Shenoy does not argue that Shappell teaches away from Sigurdsson’s express disclosures regarding these concepts. A POSA would not have been dissuaded by anything in Shappell from synchronizing files of the same user across multiple user devices as taught in Sigurdsson, particularly in view of Sigurdsson’s explanation of the obvious benefits of keeping a user’s files synchronized across multiple user devices, such as a user’s home and work computers. *See, e.g.*, Sigurdsson, ¶ 26.

132. Moreover, I disagree with Dr. Shenoy’s characterization of Shappell. The fact that group members in Shappell’s shared spaces may share files between each other necessarily means that the computers involved in such synchronization

are associated with the user in the sense that the user is part of the same group in which the files are shared and synchronized. In fact, Shappell expressly discloses predetermination of “who can be members” of the shared group using a group policy associated with the user that would be applied to “all the machines in the domain to which the policy pertains.” Shappell, ¶ 87. Thus, all machines in the particular domain would be associated with the user in the sense that they adopt the same group policy as the user’s device.

VIII. THE CHALLENGED CLAIMS WOULD HAVE BEEN OBVIOUS IN VIEW OF BROWN IN COMBINATION WITH HESSELINK

A. Summary of the Prior Art

133. I hereby incorporate by reference § VII.A of my Opening Declaration as if fully stated herein, which summarizes, *inter alia*, the Brown (Ex. 1025) and Hesselink (Ex. 1026) prior art references.

B. Claims 1, 3, 4, 8, 10 and 11 of the ’561 Patent Would Have Been Obvious in View of Brown in Combination with Hesselink

1. Independent Claim 1

- a. “the first application [executed on the first electronic device] is further configured to automatically transfer the modified first electronic file copy to the third electronic device”**

134. Dr. Shenoy opines that Brown and Hesselink do not disclose or render obvious that “the first application [executed on the first electronic device] is further configured to automatically transfer the modified first electronic file copy to the third

electronic device,” as this limitation appears in claim 1 of the ’561 patent (similar limitations appear in each of the Challenged Claims). Shenoy Decl., ¶ 94. I disagree. In my opinion, Brown in combination with Hesselink either expressly disclose or at least suggest or render obvious this claim element, for the reasons discussed below and in my Opening Declaration.

135. As discussed in my Opening Declaration, Brown discloses a “first electronic device” (server 105) that includes a “first application” (Synchronization File System (“SFS”) 110). Brown, 3:56-57. Brown also discloses a “second electronic device” and “third electronic device” in the form of, e.g., computers 130 and 135, respectively, which could be a user’s desktop and portable computers, respectively. Brown, 1:24-50, 4:20-24. I explain in detail how it would have been obvious in view of Brown in combination with Hesselink for client 130 to send a file to server 105 when the user modifies content of the file. *See* Balakrishnan Open. Decl., ¶¶ 828-35; *see also id.*, ¶¶ 601-11 (explaining why a POSA would have been motivated to combine the relevant teachings of Brown with relevant teachings from Hesselink). In addition, I explain in detail why it would have been obvious to a POSA to automatically transfer the modified file from client 130 stored at server 105 to client 135. *Id.*, ¶¶ 828-35.

136. Dr. Shenoy argues that this claim element requires the server to automatically transfer a modified file “without a request from the client for the

modified file[.]” Shenoy Decl., ¶ 94. I disagree. Nothing in the claims or specification requires transfer of the modified file from the server to the client without a request from the client for the modified file. In fact, the Challenged Patents reference “push” and “pull” approaches as if they are trivially interchangeable, and a POSA would understand from reading such disclosure that the claims do not require a server to push the modified first electronic file copy from the first electronic device to the third electronic device:

While a preferred embodiment of the invention has been illustrated and described, as noted above, many changes can be made without departing from the spirit and scope of the invention. For example, as an alternative to the approach described with reference to FIG. 3, wherein the transfer clients 310, 320 function to “push” modified or created files to the synch application 380, the synch application 380 may instead function to periodically “pull” or otherwise actively retrieve such files from the transfer clients 310, 320[.] Instead, the invention should be determined entirely by reference to the claims that follow.

’942 patent, 10:48-58. The ’942 and ’561 patents’ inventors knew how to distinguish between “push” and “pull” data transfer models based on the above disclosures but nevertheless decided not to limit the claims of the patents to require use of either one of these models by not using the word “push” (or the concept itself) in any of the Challenged Claims. Thus, Dr. Shenoy is wrong to simply assume that

a push model is required, particularly given that he cites no language from the Challenged Patents requiring such an interpretation.

137. One part of the '942 and '561 patents' specification states:

Typically, the server 230 will store one or more of the received files to the storage 270 only if the destination user device is offline or otherwise temporarily not in communication with the server 230. Upon resuming communication with the destination user device, the server 230 will transfer the temporarily stored files to the destination device.

'942 patent, 8:26-31. In my opinion, a POSA would understand this disclosure to contemplate a previously-offline user device first requesting to resume communication with the server before the server sends a modified file to the user device. Thus, Dr. Shenoy is wrong in assuming the claims require the absence of a request from a user device before a server sends a modified file to the user device.

138. The Challenged Patents expressly contemplate user devices connecting to a server via the internet. '942 patent, 7:28-32. As such, a POSA would have understood the Challenged Claims to cover the situation in which a previously-offline user device reconnects to the server via a public network over the internet using DHCP. For the reasons discussed above in § VI, the user device would have been **required** in this scenario to first reestablish communication with the server (not the other way around). Thus, Dr. Shenoy is incorrect in interpreting the Challenged Claims to require the **server** to transfer a modified file to a “third electronic device”

“without a request from the client[.]” Shenoy Decl., ¶ 94.

139. Dr. Shenoy states that “Brown discloses that communication between the clients and server uses the hypertext transport protocol (HTTP).” *Id.*, ¶ 95. However, Brown states: “Preferably, the client SA communicates with the server SFS using a proprietary protocol tunneled within the hypertext transport protocol (HTTP), ***but a person skilled in the art will recognize that other protocols can be used.***”). Brown, 3:1-8. In addition, Hesselink discloses: “The invention is described primarily in terms of use with HTTP (Hypertext Transfer Protocol), ***but may be used with other data transfer protocols.***” Hesselink, ¶ 75. Thus, Brown and Hesselink expressly contemplate the use of any conventional communication protocol, including HTTP and other protocols that would have been within the background knowledge of a POSA and supported both push and pull models of data transfer, as discussed above in § VI.

140. A POSA would have understood or at least found it obvious from these disclosures that either or both clients could automatically send files to the server using HTTP and the server could automatically send files to either or both clients. In fact, Hesselink expressly states that “data” may be embedded in HTTP ***requests*** sent from a client computer to a “connection server” discussed in Hesselink, thus expressly contemplating the concept of pushing data using HTTP. Hesselink, ¶ 87; *see also id.*, ¶ 89 (client HTTP request to “security server” includes embedded

“authentication data and message digest value”); *id.*, ¶ 94 (“In event 526, process 32 on client computer 12a-12n sends an **HTTP request** to security server 58 **that includes the encrypted secret key and corresponding message digest value embedded within the request.**”).

141. Two of Hesselink’s embodiments associated with Figs. 7 and 8 (which are flow charts that are implementable using the system shown in Fig. 2) expressly contemplate using HTTP to effectively push data from a source client to a server and then from the server to a destination client. Specifically, paragraph 105 of Hesselink describes a source client (e.g., any one of items 12a, 12b, 12c, etc. in Fig. 2) checking whether it has data to send through a server (e.g., any of items 14a, 14b, 14c, 14d, 14e, 14f, 14g, 14h, or 14n in Fig. 2) to a destination client (e.g., any one of items 18a, 18b, 18c, 18d, 18e, 18f, 18n-1, or 18n in Fig. 2). Hesselink, Fig. 7, ¶ 105. If so, the source client embeds the data in an HTTP request, such as an HTTP POST message, that is sent to a “connection server.” *Id.*, ¶ 106. The HTTP response from the connection server to the source client may push data from a **different** client to the source client. *Id.*, ¶¶ 107, 109 (the connection server “may serve as a bridging medium between client computer(s)”). Paragraph 109 of Hesselink further discloses that a source client may push data to a connection server that subsequently pushes the same data to a destination client via an HTTP response to an unrelated HTTP request from the destination client. *Id.*, ¶ 109. Whether an HTTP response or request

is used to send the data is irrelevant—the data is pushed from the server to the destination client in an HTTP response even though the destination client did not request the data from the connection server. *Id.* A POSA would understand that this is equivalent to a push—the data simply arrives at the destination client in an HTTP response associated with an unrelated HTTP request previously sent by the destination client. *Id.*; *see also* Hesselink, Fig. 8, ¶¶ 110-15 (discussing the same sort of functionality from the perspective of the connection server).

142. As such, Dr. Shenoy is incorrect in stating that “the HTTP protocol does not support push-based functionality at the time of the invention.” Shenoy Decl., ¶ 95. In addition, a POSA would have understood how to implement push-type data transfers including by using disclosures such as those in the Hepworth reference, discussed above. In view of the above disclosures in Hesselink, a POSA would have understood and found it obvious to use Brown’s server 105 to send a message to client 135 containing a file modified by the user on client 130, thereby pushing such modified file to client 135.

143. The Challenged Claims and specification do not require the use of a “push” model of data transfer or the absence of intervening steps between when a modified file is received from one user device at server 105 and when it is forwarded by server 105 to another user device. For example, claim 4 of the ’561 patent expressly requires the “first application” (on, e.g., server 105 in Brown) to “store the

modified first electronic file copy to the memory device associated with the first electronic device when the third electronic device is not in communication with the first electronic device.” The Challenged Patents describe a “synch application” that “polls” client devices to determine whether they are in communication with the server running the synch application. ’942 patent, 8:50-54. This polling mechanism is also described in Brown (3:22-26) and Hesselink (§ 109), which confirms that a POSA would not read the Challenged Claims to require the absence of any user request before an updated file is automatically sent to a client from a server. Thus, claim 4 and the Challenged Patents’ specification contemplates at least one intervening step that does not appear in claim 1 but is required before a file may be transferred to the third electronic device in claim 4—namely, determining whether the first electronic device is in communication with the third electronic device. Given that claim 4 depends (indirectly) from claim 1, it necessarily follows that claim 1 must include within its scope implementations that do not immediately forward modified files from one client device through a server to another client device but, instead, implement other steps in between, such as allowing a previously-offline client to reestablish communication with the server, request a list of missing content, and then request the missing content itself.

144. I understand from counsel that a claim is obvious if a POSA would have been motivated to combine teachings from various embodiments in a single prior art

reference based on the teachings of those embodiments, the general knowledge present in the art, and common sense. I have also been informed by counsel that a POSA has ordinary creativity, and that there is no requirement that there be an actual or explicit teaching to combine two embodiments in a single prior art reference. Dr. Shenoy does not appear to be applying these principles from the perspective of a POSA in his analysis of the Brown and Hesselink prior art combination, as discussed in more detail below.

145. The peer-to-peer embodiments of Hesselink and drop-box techniques of Brown may be employed interchangeably when sending and receiving clients are online—the sending client may either send to server 105, which can then forward the file to the receiving client, or the sending client may send to the receiving client directly. A POSA would have viewed either of these options as obvious, conventional, and well-known design choices that would have been obvious to try in view of Brown’s and Hesselink’s disclosures.

146. A POSA would understand that server 105 automatically sends the client 130 content to client 135, even though client 135 requests a list of missing content and requests the missing client 130 content from server 105. A POSA would view the sending of client 130 content from the server to client 135 to be automatic both from the server’s perspective and the user’s perspective. The intervening steps do not change the automatic nature of sending the client 130 content by the server

to client 135. As noted above, all independent claims recite “comprising,” which I understand to permit additional steps such as those involving client 135 requesting a list of missing content and requesting the missing content from the server while still practicing the Challenged Claims.

147. Dr. Shenoy appears to opine that Brown does not disclose multiple user devices synchronizing with the same server. Shenoy Decl., ¶ 97. To the contrary, Brown describes “a client/server application that allows users to synchronize data on multiple machines. The server component is a server Synchronization File System (SFS) with a functional interface and metadata organization that enables efficient and reliable synchronization by the clients. . . . The client component is a client Synchronization Application (SA) that uses the server SFS to synchronize its local client data with the server.” Brown, 2:57-65. Brown also discloses: “The server SFS supports *concurrent synchronization of a large number of clients* limited only by the server’s performance and bandwidth considerations.” Brown, 4:55-57.

148. As explained in my Opening Declaration, Brown addresses the problem in the prior art that an individual user may have multiple computers, such as work, home, and portable computers, and “needs to access the same files from the various computers”:

But as computers have become more affordable, people find themselves

working with several machines. It is increasingly common for people to find themselves working on one machine at the office, a second machine at home, and having use of a portable computer when they need to have computer access while traveling. . . .

But with the increasing number of machines a person might find himself using comes an added complexity. Since a person typically accesses the same files from the various computers, the user needs to be certain that the files he is accessing are current.

Originally, people carried files on floppy disks from one machine to the next. But the increases in file size sometimes make floppy disks impractical. And if the user forgets to bring the files with him as he moves around, or forgets to move the latest versions of the files off the computer he most recently used, the user can find himself with several versions of the files, each of which contain desired portions.

Accordingly, a need remains for a way to maintain distributed files across multiple clients, maintaining currency at each client as changes are made, to address these and other problems associated with the prior art.

Brown, 1:24-50. Consistent with these teachings, Figure 1 specifically discloses two client computers, client 130 which is a user's desktop computer, and client 135 which is the user's portable computer: "Client 130 represents an ordinary desktop computer, as might be present at a user's place of work or home. Portable computer

135 represents a notebook or laptop computer, as a user might take with him to a hotel room on business.” *Id.*, 4:20-24. In view of Brown’s teachings regarding the problem to be solved, a POSA would have understood that computers 130 and 135 are both, and obviously would both be, associated with the same user. Brown contemplates “multiple clients of a single user account” that may “push changes to the server concurrently” and that “the server SFS interleaves the changes.” *Id.*, 4:62-64. Therefore, “each user can have several clients accessing a single folder.” *Id.*, 11:55-56. Thus, Brown discloses and renders obvious multiple client devices associated with a single user and synchronizing with server 105.

149. A POSA would have found it obvious in view of the foregoing disclosures to automatically send a file modified on client 130 and stored at server 105 to client 135, and the fact that client 135 must initiate the synchronization process does not mean that the process is not “automatic” from the user’s and server’s perspective. A POSA would understand from Brown’s disclosures that each client implements a “sync poll call to the server” to initiate the synchronization process periodically. Brown, 5:14-17. A POSA would have understood “poll” to mean “[t]he process of periodically determining the status of each device in a set so that the active program can process the events generated by each device[.]” Ex. 1040, Microsoft Computer Dictionary (5th ed. 2002) (definitions for “polling” and “autopolling”). Thus, a POSA would have understood computers 130 and 135 to

periodically check with server 105 for updates as part of their respective, ongoing synchronization processes. Brown, 5:14-17. A POSA would understand this polling process to automatically cause the server to synchronize with each client, including computers 130 and 135, causing an updated file from client 130 to be sent through server 105 to client 135. Indeed, the Challenged Patents use the same polling mechanism for synchronization. *See* '942 patent, 8:51-54.

150. Dr. Shenoy opines: “Clearly, Brown uses a protocol that uses a ‘pull’ mechanism and not the server automatically pushing the modified file to the client.” Shenoy Decl., ¶ 100. However, Dr. Shenoy’s own cited evidence confirms that a POSA would have found it obvious in view of Brown and Hesselink to use Brown’s server 105 to implement push-based data/file transfers to clients, because a POSA would know that HTTP POST messages, as described in Hesselink (as discussed above), or HTTP PUT messages may be used to effectively push data to a device receiving the HTTP POST or PUT messages. IETF HTTP/1.1, § 9.5 (“The POST method is used to request that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line.”); *id.*, § 9.6 (“The PUT method requests that the enclosed entity be stored under the supplied Request-URI.”).

151. Moreover, the Challenged Claims do not recite the word “push” and do not require a push mechanism to transfer data from the server to any client. The

inventors knew how to use the words “push” and “pull” to describe data transfers and purposely did not limit the Challenged Claims to being implemented via push or pull.

152. Thus, the fact that Brown may use a client-pull mechanism to *initiate* synchronization with server 105 and Dr. Shenoy’s arguments that attempt to mischaracterize Brown in combination with Hesselink as exclusively teaching, suggesting, and rendering obvious embodiments that use pull mechanisms are ultimately irrelevant. There is nothing in the Challenged Patents or their file histories, such as a clear disclaimer or disavowal of claim scope or lexicography, that could justify limiting the claims to cover embodiments in which no client requests are sent to Brown’s server. In fact, the Challenged Patents teach the opposite: a client *must* respond to a server’s polling request to allow the server to first determine that the client is in communication with the server. ’942 patent, 8:45-59. Even though the Challenged Claims do not expressly recite using such a client request, the Challenged Claims use the word “comprising,” which means that Brown in combination with Hesselink disclose and render obvious the Challenged Claims even though they may contemplate certain automated communications between clients and the server that are not expressly recited in the Challenged Claims.

153. Even if the Board were to conclude that one or all of Brown’s embodiments, as modified in view of Hesselink, are limited to pull-based HTTP

mechanisms as Dr. Shenoy incorrectly asserts, it would have been obvious in view of the Hesselink embodiments discussed above disclosed in connection with paragraphs 104-115 of Hesselink to push data from server 105 to a client. Use of such functionality would have been in the background knowledge of the POSA, as demonstrated by the fact that, e.g., such paragraphs disclose the use of such HTTP messages without even describing what they are. *See* Hesselink, ¶¶ 104-15. The embodiments describe pushing data from server 105 to clients without the need to go into any detail about HTTP or other well-known file transfer methods that could have been used to implement Brown's and Hesselink's systems and methods that implement such functionality. There should be no dispute that such push-based functionality was in the background knowledge of a POSA, given that Dr. Shenoy himself relies upon evidence that confirms this understanding. *See* IETF HTTP/1.1, §§ 9.5, 9.6.

154. In addition, neither Brown nor Hesselink *requires* use of HTTP/HTTPS, and a POSA would have known how to implement push-based file synchronization using push-based applications. The Challenged Patents refer to “a browser *or other applications*, running on one or more of the user devices 210, 280 [that] can cooperate in two-way communication with server 230 and one or more applications running on server 230.” ’942 patent, 7:51-54. The Challenged Patents provide no detail about how to implement any of these applications, confirming that

doing so would have been routine, obvious, well-known, and conventional prior to November 2008. Indeed, there were many well-known, conventional examples of applications used to push data from a server to a user's device that were in public use prior to November 2008, as discussed above in § VI with respect to the Wong reference.

155. Moreover, Brown expressly states: “From the user’s perspective, the synchronization process is automatic, runs in the background and requires minimal monitoring or intervention.” Brown, 3:40-42. As Dr. Shenoy himself explains, the Challenged Claims are “user-centric[.]” Shenoy Decl., ¶ 22. Thus, all that is required is that the synchronization process be automatic *from the user’s perspective*. Dr. Shenoy’s own statement in paragraph 22 of his declaration therefore contradicts his later statement in paragraph 102 that the transfer of the modified file from server to client 135 must be automatic from the *server’s* perspective. *Id.*, ¶¶ 22, 102.

156. Dr. Shenoy incorrectly states: “While Brown’s file synchronization requires the client to execute prescribed steps, Brown expressly teaches its ‘automatic synchronization’ as being automatic from the *user’s* perspective as long as the client does not necessitate any action from the user in order to initiate the synchronization.” Shenoy Decl., ¶ 103 (emphasis in original). In my opinion, this statement by Dr. Shenoy misrepresents the disclosures in Brown. Brown does not

state or even suggest *anywhere* in its disclosures that *any* “action from the user” is necessary “in order to initiate the synchronization.” *Id.* Indeed, Dr. Shenoy fails to identify any such disclosure in Brown in his declaration.

157. I disagree with Dr. Shenoy’s statement that “the automatic transfer in the claims of the ’561 Patent is automatic from the *server’s* perspective[.]” Shenoy Decl., ¶ 104 (emphasis in original). To the contrary, claim 4 of the ’561 patent recites: “the first application [running on server 105] is configured to store the modified first electronic file copy to the memory device associated with the first electronic device [server 105] when the third electronic device [client 135] is not in communication with the first electronic device [server 105].” Claim 4 depends indirectly from claim 1, and similar language in claim 4 appears in claim 11, which depends indirectly from claim 8. In order for claims 4 and 11 to make any sense in the context of their respective independent claims, the independent claims must be read to *not* require synchronization to be automatic “from the server’s perspective,” because the server has absolutely no control whatsoever over whether or when any particular client may go offline or whether or when it may attempt to reestablish communications with the server. In other words, in at least the embodiments claimed in claims 4 and 11, the server *cannot* automatically send any file to an offline client without the offline client first coming back online, reestablishing communication with the server, and requesting the file (whether it be as part of—or

as a result of—the reestablishment of communication or otherwise).

b. A POSA Would Have Been Motivated to Look to Hesselink in View of Brown and Would Have Found It Obvious in View of Brown and Hesselink to Automatically Send a Modified File from Brown’s Client 130 to Server 105 and Then to Client 135

158. Dr. Shenoy states: “A POSA would not rely on Hesselink at ¶ 183 as disclosing teaching automatic push from the server.” Shenoy Decl., ¶ 107. I disagree. As discussed above, Hesselink discloses in paragraphs 104-115 embodiments that are strikingly similar to those described in Brown, and Hesselink further discloses “when data is updated, such as when a file is edited, for example, the system may be configured to automatically update all or some designated subset of computers on the network as soon as the update has been performed at one location.” Hesselink, ¶ 183. A POSA would understand this statement in paragraph 183 of Hesselink to refer to the “system” as a whole, including the embodiments depicted in Figs. 7 and 8 and described in paragraphs 104-115. Dr. Shenoy is therefore incorrect in stating: “Hesselink at the cited portion or anywhere else does not contain any disclosure of the server automatically pushing the modified file.” Shenoy Decl., ¶ 107. To the contrary, as discussed above, the embodiments in paragraphs 104-115 expressly disclose automatically pushing modified files from a source client to a connection server and then from the connection server to a destination client. Hesselink, ¶¶ 104-15. Paragraph 183 only reinforces the idea that

it would have been obvious in view of Brown in combination with Hesselink to automatically send an updated file from a source client to a server and from the server to a destination client “as soon as [an] update [on the file] has been performed[.]” *Id.*, ¶ 183.

159. Dr. Shenoy opines: “Hesselink only discusses the pull-based synchronization throughout its disclosure, for example, as illustrated in detail with reference to FIGS. 4 and 8.” Shenoy Decl., ¶ 108. I disagree. Like the embodiments associated with Figs. 7 and 8 of Hesselink discussed above, the embodiment associated with Fig. 4 of Hesselink expressly discloses pushing “data” from a source client through a connection server to a destination client. For example, step 450 described in paragraph 87 of Hesselink involves the sending of data from a source client (e.g., any one of items 18a, 18b, 18c, 18d, 18e, 18f, 18n-1, or 18n in Fig. 2) through a server (e.g., any of items 14a, 14b, 14c, 14d, 14e, 14f, 14g, 14h, or 14n in Fig. 2) to a destination client (e.g., any one of items 12a, 12b, 12c, etc. in Fig. 2). Hesselink, ¶ 87. Hesselink explains that “full duplex connections” between the “12” clients and the “18” clients may be established, and a POSA would understand or have found it obvious in view of these disclosures that a modified file from a source client would have been sent to a connection server and then forwarded automatically to a destination client. *Id.*

160. Contrary to Dr. Shenoy’s assertions, this embodiment associated with

Fig. 4 in Hesselink does **not** use a “pull mechanism[.]” Shenoy Decl., ¶ 110. Instead, paragraph 87 of Hesselink clearly describes a **push** mechanism whereby data from one or more of the source clients (designated as items with labels beginning with “18” in Fig. 2) push data through a connection server to a destination client (designated as items with labels beginning with “12” in Fig. 2). Hesselink, ¶ 87. After a connection is established, there is no requirement for a client to initiate any request for any update to or from the connection server, contrary to Dr. Shenoy’s assertions—content is merely pushed to the client unsolicited in HTTP responses.

161. Likewise, as discussed, Hesselink’s embodiments depicted in Figs. 7 and 8 involve the use of a push mechanism to send data from a source client, through a server, to a destination client automatically. Therefore, I also disagree with Dr. Shenoy’s assertions that the embodiment associated with Fig. 8 in Hesselink is “focused on a data transfer process based on client pull.” Shenoy Decl., ¶ 110.

162. In addition, as discussed above, I disagree with Dr. Shenoy’s assertion (which he repeats in paragraph 111 of his declaration) that a POSA would have only known how to use HTTP to pull data from a server, not push data. *Id.*, ¶ 111. As discussed above in § VI, a POSA would have understood and found it obvious in view of references like Hepworth and Dr. Shenoy’s own cited references regarding HTTP that HTTP could be used to implement data pushes from a server. Moreover, Hesselink does not even require the use of HTTP, and a POSA would have found it

obvious to use any available communications protocol to implement data pushes from the “connection servers” discussed in Hesselink to destination clients. Dr. Shenoy himself quotes excerpts from Hesselink that disclose a connection server 14 pushing data from “network-enabled device” 18 to “client computers 12[.]” Shenoy Decl., ¶ 111 (quoting Hesselink, ¶ 78 (“client computers 12 a, 12 b may each comprise programming 32 stored in memory that allow client computers 12 a, 12 b to send instructions to network-enabled devices 18 a, 18 b as requests through connection server 14, ***and receive data from network-enabled devices 18 a, 18 b via responses through connection server 14***”)). As discussed above, the fact that a client request may initiate such synchronization processes is irrelevant.

163. Dr. Shenoy discusses yet another example from Hesselink that uses a server to push a modified file from one client to another client. Shenoy Decl., ¶ 112 (quoting Hesselink, ¶ 141). The embodiment discussed in paragraph 141 of Hesselink is depicted in Fig. 10, which, again, shows multiple clients (72, 74, 76) all connected to a “connection server” (“14”). See Hesselink, Fig. 10, ¶ 141. Paragraph 141 explains that one client (72) may send a file modification to another client (74), which in turn may broadcast the file modification to all other “subscribed user modules.” *Id.*, ¶ 141. Such pushing of file changes to “subscribed” clients, when implemented using the system in Fig. 10, would necessarily make use of, or at least render obvious the use of, the connection server depicted in Fig. 10, thereby

necessarily making use of a push-based transfer of content from, e.g., client 74 through a connection server to the destination client, e.g., client 76 receiving subscribed file modifications. *Id.*, Fig. 10. The fact that in some embodiments an initial request from a client must be made to establish an active subscription (*see id.*, ¶ 140) does not change the fact that all subsequent data transfers from one client through the connection server to a destination client are performed using a push-based model from the server to the destination client—in fact, Dr. Shenoy does not appear to dispute this point in connection with the embodiments discussed in paragraphs 140-141 in Hesselink. *See* Shenoy Decl., ¶ 112. The initial request to subscribe does not change the fact that all subsequent data transfers are automatic, consistent with what is claimed in the Challenged Claims.

c. Hesselink Does Not Teach Away from Brown's Disclosures Related to Synchronizing Using a Server and Two Devices All Associated with the Same User

164. As discussed in my Opening Declaration, Brown expressly discloses two user devices (computers 130 and 135) and a server (105) that are associated with the same user. *See* Balakrishnan Open. Decl., ¶¶ 585-87. Dr. Shenoy does not appear to dispute these opinions anywhere in his declaration. Instead, he asserts: “Hesselink fails to disclose the first application associated with a user automatically transferring a modified first electronic file copy to the third electronic associated with a same user.” Shenoy Decl., ¶ 114. I have been informed by counsel that not

every limitation needs to be disclosed in every prior art reference relied upon in an obviousness combination. Based on this principle, it is irrelevant whether Hesselink discloses clients and a server associated with the same user, because I am relying on Brown's teachings for this limitation, not Hesselink's.

165. Nevertheless, Hesselink does not teach away from synchronizing using a server and multiple client devices associated with the same user. For example, Hesselink teaches in paragraph 84 that the connection servers (14) shown in Fig. 2 may be operatively coupled to a security server 58 shown in Fig. 3. Hesselink, ¶ 84. In turn, the security server uses an "authentication application" to "see whether or not the authentication data in the request" from a client computer (12) "verifies the *user* of client computer 12" such that the client computer is assigned a connection server (14), as discussed in paragraph 86 of Hesselink. *Id.*, ¶¶ 85-86. Thus, Hesselink expressly discloses association of a particular user to both client computer 12 and an "assigned" connection server 14. *Id.*, ¶ 86. Subsequently, a connection is established between *the same* connection server 14 that was "assigned" to client computer 12 and a "network-enabled device" 18 "*selected by the user.*" *Id.* Thus, both clients 12 and 18 and the "assigned" server 14 are all expressly described as being associated with *the same* user in Hesselink.

166. Dr. Shenoy is thus incorrect in suggesting that all client computers 12 may be connected to all network-enabled devices 18. Shenoy Decl., ¶ 119 (citing

Hesselink, ¶¶ 76, 120-21). This is not what Hesselink teaches. Dr. Shenoy cites paragraph 120 of Hesselink (*see* Shenoy Decl., ¶ 119), which states: “The present invention enables a user to access devices 18a-18n of all the computers ***that a user would normally have access to***, from any single location throughout the world where the user has network access, assuming all of the computers also have network access.” Hesselink, ¶ 120. This expressly teaches that the user only has access to computers “that a user would normally have access to,” like a work computer—in other words, computers that are associated with the user. This is consistent with the disclosures in connection with Fig. 10 and paragraph 132 of Hesselink cited by Dr. Shenoy, which merely confirm the customizability of the computers to which a user may have access. Shenoy Decl., ¶ 120 (citing Hesselink, Fig. 10, ¶ 132).

167. Dr. Shenoy is simply wrong in stating that “Hesselink does not make a determination as to whether the third electronic device is associated with the same user by use of the first, second, and third applications recited in the asserted claims, and hence Hesselink cannot make an automatic transfer of a modified file based on such determination.” Shenoy Decl., ¶ 120. As discussed above, Hesselink, expressly teaches deciding, at the outset, whether the first, second, and third devices (i.e., client 12, connection server 14, and network-enabled device 18) are authorized to communicate with each other and, if so, thereby confirming all three devices must necessarily be associated with the same user.

168. Dr. Shenoy cites paragraph 127 of Hesselink and opines: “Hesselink does not teach a dedicated file transfer and/or synchronization application as recited in the asserted claims.” Shenoy Decl., ¶ 120. Petitioners and I rely upon the applications discussed in Brown for this claim limitation, not Hesselink. Moreover, such applications *are* disclosed in Hesselink in, e.g., the form of software modules 72su and 74su as discussed in paragraph 127 (and other modules and applications discussed elsewhere in Hesselink). By virtue of the fact that these applications are associated with devices that must have been authenticated by the security server, as discussed above, in order to communicate with each other, they are necessarily associated with the same user.

d. A POSA Would Have Been Motivated to Combine Brown and Hesselink and Found the Challenged Claims Obvious in View of Such Combination

169. As discussed above, I do not believe that the Challenged Claims require either a “push”- or “pull”-based model for file transfers from, e.g., the “first electronic device” to the “third electronic device” in claim 1 of the ’561 patent. The inventors knew how to distinguish between these concepts and nevertheless chose not to specify one or the other (or both) in the Challenged Claims.

170. Dr. Shenoy states: “a POSA would not be motivated to modify Brown’s disclosure to that of a push-based file transfer due to the differences between the pull and push-based methods.” Shenoy Decl., ¶ 122. Even if a push model were required

to practice the claims (it is not), I disagree with this statement from Dr. Shenoy. Brown expressly contemplates the concept of pushing changes using server 105's SFS application to syncing clients: "The server SFS uses this value to quickly find directories with changes that need to be *pushed down to a syncing client*. Directory items with DSI values greater than the value passed in by the polling client SA identify the directories with changes." Brown, 5:36-40. In addition, for the reasons discussed above, push-based messaging would have been well-known and within the background knowledge of a POSA. Thus, it would have been obvious to a POSA that such push-based messaging could be used in connection with Brown's embodiments, particularly in view of Hesselink's teachings, which expressly teach push-based messaging, as discussed in my Opening Declaration and above. Balakrishnan Open. Decl., ¶¶ 807-35.

171. Dr. Shenoy quotes Brown's desire to "minimize the load on the server processor[.]" Shenoy Decl., ¶ 124 (quoting Brown, 3:17-28). This desire would have motivated a POSA to look to a reference like Hesselink for teachings related to how to synchronize files as quickly as possible by immediately and automatically sending modified files received from one client computer associated with a user to another client computer associated with the same user using HTTP responses as taught in Hesselink. Pushing modified files as quickly as possible using, e.g., Hesselink's "subscription" method discussed above, would minimize the burden on

Brown's server. Thus, I disagree with Dr. Shenoy that "the intended principle of operation and purpose of Brown teaches away from using the server-push mechanism for file synchronization[,]" Shenoy Decl., ¶ 125, and I disagree with his statement that using push-based messaging would "fundamentally change the operation of Brown's system[,]" *id.*, ¶ 126. Instead, a POSA would have recognized push-based messaging as an obvious alternative that could be used in Brown's system in view of Hesselink's teachings, as discussed in my Opening Declaration. Balakrishnan Open. Decl., ¶¶ 807-35.

172. Dr. Shenoy references Hesselink's "peer-to-peer embodiment" associated with Fig. 11 of Hesselink and argues that it somehow teaches away from combining teachings of Brown and Hesselink. Shenoy Decl., ¶¶ 127-35. I disagree. First, I disagree with Dr. Shenoy's statement that the embodiment described in paragraph 183 of Hesselink is solely related to the embodiments associated with Figs. 11-13 in Hesselink. *Id.*, ¶ 128. As discussed above, the embodiment associated with Fig. 10 includes a connection server, and paragraph 183 of Hesselink applies to that embodiment to at least the same extent it applies to the embodiments in Figs. 11-13.

173. Like Sigurdsson, Hesselink describes peer-to-peer and client-server architectures as being well-known, conventional, and obvious variants for implementing the systems and methods of Hesselink. *See, e.g.*, Hesselink, ¶ 126

(“Computers 72 and 74 are connectable across a wide area network by any of the means described above with regard to FIG. 10, for example, via a connection server 14, or directly connected, with or without a VPN.”). Thus, a POSA would not have viewed Hesselink as teaching away from Brown’s client-server-based systems and methods—as discussed above, Hesselink itself discloses multiple embodiments that are based on a client-server architecture. Thus, it is not solely based on a peer-to-peer architecture.

174. Contrary to Dr. Shenoy’s assertions (*see* Shenoy Decl., ¶ 132), the security concerns in paragraph 6 of Hesselink are not relegated solely to peer-to-peer implementations of file sharing. *See* Hesselink, ¶ 6 (“While it is possible to upload digitized photos to a web page provided by a photo service center, this solution leave some things to be desired too, since many users may be uncomfortable storing photos of their families on some photo service server in some unknown location where it is also often unknown what level of security (if any) such server is provided with.”). Instead, these concerns are addressed by Hesselink’s security server, as discussed above, and the security server is expressly disclosed in Hesselink as being used to set up a client-server architecture for file sharing, as discussed above.

175. Dr. Shenoy states: “while Hesselink mentions including a server within its peer-to-peer system, such server only serves as a pipeline or conduit between two peers.” Shenoy Decl., ¶ 133. But Dr. Shenoy does not describe what, if anything,

more is required of the server aside from effectively acting as a pipeline or conduit between two peers. A POSA would understand any transfer of a file from a source client to a server and then from a server to a destination client to necessarily involve some storage at the server, even if only temporarily. And Dr. Shenoy does not argue that anything in Hesselink teaches away from Brown's express teachings related to the server storing a file from a source client before sending it on to a destination client, as discussed in my Opening Declaration. Balakrishnan Open. Decl., ¶¶ 848-62.

176. I disagree with the Patent Owner's assertion in its Patent Owner's Response (IPR2023-00427, Paper No. 30) ("POR")⁴ that paragraphs 6 and 13 in Hesselink describe "specific problems arising in the client-server system that is adopted in Brown." POR, 61. To the contrary, problems of security, difficulty in installation, cost, and maintenance (*see* POR, 61) would have arisen in **both** peer-to-peer **and** client-server implementations.

⁴ I understand that a substantively identical argument was presented in the Patent Owner Responses for both the IPR involving the '561 patent and the IPR involving the '942 patent.

C. Claims 1, 3, 4, 10, 12 and 13 of the '942 Patent Would Have Been Obvious in View of Brown in Combination with Hesselink

1. Independent Claim 1

- a. “automatically send[ing], via the first application, the modified first electronic file copy to a third application at the third electronic device . . . responsive to receiving the modified first electronic file copy from the second electronic device”**

177. Dr. Shenoy opines that Brown and Hesselink do not disclose or render obvious “automatically send[ing], via the first application, the modified first electronic file copy to a third application at the third electronic device . . . responsive to receiving the modified first electronic file copy from the second electronic device” as this limitation appears in claim 1 of the '942 patent (similar limitations appear in each of the Challenged Claims). Shenoy Decl., ¶ 191. I disagree. In my opinion, Brown in combination with Hesselink either expressly disclose or at least suggest or render obvious this claim element, for the reasons discussed below and in my Opening Declaration.

178. As discussed in my Opening Declaration, Brown discloses a “first electronic device” (server 105) that includes a “first application” (Synchronization File System (“SFS”) 110). Brown, 3:56-57. Brown also discloses a “second electronic device” and “third electronic device” in the form of, e.g., computers 130 and 135, respectively, which could be a user’s desktop and portable computers, respectively. Brown, 1:24-50, 4:20-24. I explain in detail how it would have been

obvious in view of Brown in combination with Hesselink for client 130 to send a file to server 105 when the user modifies content of the file. *See* Balakrishnan Open. Decl., ¶¶ 828-35; *see also id.*, ¶¶ 601-11 (explaining why a POSA would have been motivated to combine the relevant teachings of Brown with relevant teachings from Hesselink). In addition, I explain in detail why it would have been obvious to a POSA to automatically transfer the modified file from client 130 stored at server 105 to client 135. *Id.*, ¶¶ 828-35.

179. Dr. Shenoy argues that this claim element requires the server to automatically transfer a modified file “without a request from the client for the modified file[.]” Shenoy Decl., ¶ 191. I disagree. Nothing in the claims or specification requires transfer of the modified file from the server to the client without a request from the client for the modified file. In fact, the Challenged Patents reference “push” and “pull” approaches as if they are trivially interchangeable, and a POSA would understand from reading such disclosure that the claims do not require a server to push the modified first electronic file copy from the first electronic device to the third electronic device:

While a preferred embodiment of the invention has been illustrated and described, as noted above, many changes can be made without departing from the spirit and scope of the invention. For example, as an alternative to the approach described with reference to FIG. 3, wherein the transfer clients 310, 320 function to “push” modified or

created files to the synch application 380, the synch application 380 may instead function to periodically “pull” or otherwise actively retrieve such files from the transfer clients 310, 320[.] Instead, the invention should be determined entirely by reference to the claims that follow.

’942 patent, 10:48-58. The ’942 and ’561 patents’ inventors knew how to distinguish between “push” and “pull” data transfer models based on the above disclosures but nevertheless decided not to limit the claims of the patents to require use of either one of these models by not using the word “push” (or the concept itself) in any of the Challenged Claims. Thus, Dr. Shenoy is wrong to simply assume that a push model is required, particularly given that he cites no language from the Challenged Patents requiring such an interpretation.

180. One part of the ’942 and ’561 patents’ specification states:

Typically, the server 230 will store one or more of the received files to the storage 270 only if the destination user device is offline or otherwise temporarily not in communication with the server 230. Upon resuming communication with the destination user device, the server 230 will transfer the temporarily stored files to the destination device.

’942 patent, 8:26-31. In my opinion, a POSA would understand this disclosure to contemplate a previously-offline user device first requesting to resume communication with the server before the server sends a modified file to the user device. Thus, Dr. Shenoy is wrong in assuming the claims require the absence of a

request from a user device before a server sends a modified file to the user device.

181. The Challenged Patents expressly contemplate user devices connecting to a server via the internet. '942 patent, 7:28-32. As such, a POSA would have understood the Challenged Claims to cover the situation in which a previously-offline user device reconnects to the server via a public network over the internet using DHCP. For the reasons discussed above in § VI, the user device would have been **required** in this scenario to first reestablish communication with the server (not the other way around). Thus, Dr. Shenoy is incorrect in interpreting the Challenged Claims to require the **server** to transfer a modified file to a third electronic device “without a request from the client[.]” Shenoy Decl., ¶ 191.

182. Dr. Shenoy states that “Brown discloses that communication between the clients and server uses the hypertext transport protocol (HTTP).” Shenoy Decl., ¶ 192. However, Brown states: “Preferably, the client SA communicates with the server SFS using a proprietary protocol tunneled within the hypertext transport protocol (HTTP), **but a person skilled in the art will recognize that other protocols can be used.**”). Brown, 3:1-8. In addition, Hesselink discloses: “The invention is described primarily in terms of use with HTTP (Hypertext Transfer Protocol), **but may be used with other data transfer protocols.**” Hesselink, ¶ 75. Thus, Brown and Hesselink expressly contemplate the use of any conventional communication protocol, including HTTP and other protocols that would have been within the

background knowledge of a POSA and supported both push and pull models of data transfer, as discussed above in § VI.

183. A POSA would have understood or at least found it obvious from these disclosures that either or both clients could automatically send files to the server using HTTP and the server could automatically send files to either or both clients. In fact, Hesselink expressly states that “data” may be embedded in HTTP *requests* sent from a client computer to a “connection server” discussed in Hesselink, thus expressly contemplating the concept of pushing data using HTTP. Hesselink, ¶ 87; *see also id.*, ¶ 89 (client HTTP request to “security server” includes embedded “authentication data and message digest value”); *id.*, ¶ 94 (“In event 526, process 32 on client computer 12a-12n sends an **HTTP request** to security server 58 **that includes the encrypted secret key and corresponding message digest value embedded within the request.**”).

184. Two of Hesselink’s embodiments associated with Figs. 7 and 8 (which are flow charts that are implementable using the system shown in Fig. 2) expressly contemplate using HTTP to effectively push data from a source client to a server and then from the server to a destination client. Specifically, paragraph 105 of Hesselink describes a source client (e.g., any one of items 12a, 12b, 12c, etc. in Fig. 2) checking whether it has data to send through a server (e.g., any of items 14a, 14b, 14c, 14d, 14e, 14f, 14g, 14h, or 14n in Fig. 2) to a destination client (e.g., any one of items

18a, 18b, 18c, 18d, 18e, 18f, 18n-1, or 18n in Fig. 2). Hesselink, Fig. 7, ¶ 105. If so, the source client embeds the data in an HTTP request, such as an HTTP POST message, that is sent to a “connection server.” *Id.*, ¶ 106. The HTTP response from the connection server to the source client may push data from a *different* client to the source client. *Id.*, ¶¶ 107, 109 (the connection server “may serve as a bridging medium between client computer(s)”). Paragraph 109 of Hesselink further discloses that a source client may push data to a connection server that subsequently pushes the same data to a destination client via an HTTP response to an unrelated HTTP request from the destination client. *Id.*, ¶ 109. Whether an HTTP response or request is used to send the data is irrelevant—the data is pushed from the server to the destination client in an HTTP response even though the destination client did not request the data from the connection server. *Id.* A POSA would understand that this is equivalent to a push—the data simply arrives at the destination client in an HTTP response associated with an unrelated HTTP request previously sent by the destination client. *Id.*; *see also* Hesselink, Fig. 8, ¶¶ 110-15 (discussing the same sort of functionality from the perspective of the connection server).

185. As such, Dr. Shenoy is incorrect in stating that “the HTTP protocol does not support push-based functionality at the time of the invention.” Shenoy Decl., ¶ 192. In addition, a POSA would have understood how to implement push-type data transfers including by using disclosures such as those in the Hepworth

reference, discussed above. In view of the above disclosures in Hesselink, a POSA would have understood and found it obvious to use Brown's server 105 to send a message to client 135 containing a file modified by the user on client 130, thereby pushing such modified file to client 135.

186. The Challenged Claims and specification do not require the use of a “push” model of data transfer or the absence of intervening steps between when a modified file is received from one user device at server 105 and when it is forwarded by server 105 to another user device. For example, claim 4 of the '561 patent expressly requires the “first application” (on, e.g., server 105 in Brown) to “store the modified first electronic file copy to the memory device associated with the first electronic device when the third electronic device is not in communication with the first electronic device.” The Challenged Patents describe a “synch application” that “polls” client devices to determine whether they are in communication with the server running the synch application. '942 patent, 8:50-54. This polling mechanism is also described in Brown (3:22-26) and Hesselink (§ 109), which confirms that a POSA would not read the Challenged Claims to require the absence of any user request before an updated file is automatically sent to a client from a server. Thus, claim 4 and the Challenged Patents' specification contemplates at least one intervening step that does not appear in claim 1 but is required before a file may be transferred to the third electronic device in claim 4—namely, determining whether

the first electronic device is in communication with the third electronic device. Given that claim 4 depends (indirectly) from claim 1, it necessarily follows that claim 1 must include within its scope implementations that do not immediately forward modified files from one client device through a server to another client device but, instead, implement other steps in between, such as allowing a previously-offline client to reestablish communication with the server, request a list of missing content, and then request the missing content itself.

187. I understand from counsel that a claim is obvious if a POSA would have been motivated to combine teachings from various embodiments in a single prior art reference based on the teachings of those embodiments, the general knowledge present in the art, and common sense. I have also been informed by counsel that a POSA has ordinary creativity, and that there is no requirement that there be an actual or explicit teaching to combine two embodiments in a single prior art reference. Dr. Shenoy does not appear to be applying these principles from the perspective of a POSA in his analysis of the Brown and Hesselink prior art combination, as discussed in more detail below.

188. The peer-to-peer embodiments of Hesselink and drop-box techniques of Brown may be employed interchangeably when sending and receiving clients are online—the sending client may either send to server 105, which can then forward the file to the receiving client, or the sending client may send to the receiving client

directly. A POSA would have viewed either of these options as obvious, conventional, and well-known design choices that would have been obvious to try in view of Brown's and Hesselink's disclosures.

189. A POSA would understand that server 105 automatically sends the client 130 content to client 135, even though client 135 requests a list of missing content and requests the missing client 130 content from server 105. A POSA would view the sending of client 130 content from the server to client 135 to be automatic both from the server's perspective and the user's perspective. The intervening steps do not change the automatic nature of sending the client 130 content by the server to client 135. As noted above, all independent claims recite "comprising," which I understand to permit additional steps such as those involving client 135 requesting a list of missing content and requesting the missing content from the server while still practicing the Challenged Claims.

190. Dr. Shenoy appears to opine that Brown does not disclose multiple user devices synchronizing with the same server. Shenoy Decl., ¶ 194. To the contrary, Brown describes "a client/server application that allows users to synchronize data on multiple machines. The server component is a server Synchronization File System (SFS) with a functional interface and metadata organization that enables efficient and reliable synchronization by the clients. . . . The client component is a client Synchronization Application (SA) that uses the server SFS to synchronize its

local client data with the server.” Brown, 2:57-65. Brown also discloses: “The server SFS supports *concurrent synchronization of a large number of clients* limited only by the server’s performance and bandwidth considerations.” Brown, 4:55-57.

191. As explained in my Opening Declaration, Brown addresses the problem in the prior art that an individual user may have multiple computers, such as work, home, and portable computers, and “needs to access the same files from the various computers”:

But as computers have become more affordable, people find themselves working with several machines. It is increasingly common for people to find themselves working on one machine at the office, a second machine at home, and having use of a portable computer when they need to have computer access while traveling. . . .

But with the increasing number of machines a person might find himself using comes an added complexity. Since a person typically accesses the same files from the various computers, the user needs to be certain that the files he is accessing are current.

Originally, people carried files on floppy disks from one machine to the next. But the increases in file size sometimes make floppy disks impractical. And if the user forgets to bring the files with him as he moves around, or forgets to move the latest versions of the files off the

computer he most recently used, the user can find himself with several versions of the files, each of which contain desired portions.

Accordingly, a need remains for a way to maintain distributed files across multiple clients, maintaining currency at each client as changes are made, to address these and other problems associated with the prior art.

Brown, 1:24-50. Consistent with these teachings, Figure 1 specifically discloses two client computers, client 130 which is a user's desktop computer, and client 135 which is the user's portable computer: "Client 130 represents an ordinary desktop computer, as might be present at a user's place of work or home. Portable computer 135 represents a notebook or laptop computer, as a user might take with him to a hotel room on business." *Id.*, 4:20-24. In view of Brown's teachings regarding the problem to be solved, a POSA would have understood that computers 130 and 135 are both, and obviously would both be, associated with the same user. Brown contemplates "multiple clients of a single user account" that may "push changes to the server concurrently" and that "the server SFS interleaves the changes." *Id.*, 4:62-64. Therefore, "each user can have several clients accessing a single folder." *Id.*, 11:55-56. Thus, Brown discloses and renders obvious multiple client devices associated with a single user and synchronizing with server 105.

192. A POSA would have found it obvious in view of the foregoing

disclosures to automatically send a file modified on client 130 and stored at server 105 to client 135, and the fact that client 135 must initiate the synchronization process does not mean that the process is not “automatic” from the user’s and server’s perspective. A POSA would understand from Brown’s disclosures that each client implements a “sync poll call to the server” to initiate the synchronization process periodically. Brown, 5:14-17. A POSA would have understood “poll” to mean “[t]he process of periodically determining the status of each device in a set so that the active program can process the events generated by each device[.]” Ex. 1040, Microsoft Computer Dictionary (5th ed. 2002) (definitions for “polling” and “autopolling”). Thus, a POSA would have understood computers 130 and 135 to periodically check with server 105 for updates as part of their respective, ongoing synchronization processes. Brown, 5:14-17. A POSA would understand this polling process to automatically cause the server to synchronize with each client, including computers 130 and 135, causing an updated file from client 130 to be sent through server 105 to client 135. Indeed, the Challenged Patents use the same polling mechanism for synchronization. *See* ’942 patent, 8:51-54.

193. Dr. Shenoy opines: “Clearly, Brown uses a protocol that uses a ‘pull’ mechanism and not the server automatically pushing the modified file to the client.” Shenoy Decl., ¶ 197. However, Dr. Shenoy’s own cited evidence confirms that a POSA would have found it obvious in view of Brown and Hesselink to use Brown’s

server 105 to implement push-based data/file transfers to clients, because a POSA would know that HTTP POST messages, as described in Hesselink (as discussed above), or HTTP PUT messages may be used to effectively push data to a device receiving the HTTP POST or PUT messages. IETF HTTP/1.1, § 9.5 (“The POST method is used to request that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line.”); *id.*, § 9.6 (“The PUT method requests that the enclosed entity be stored under the supplied Request-URI.”).

194. Moreover, the Challenged Claims do not recite the word “push” and do not require a push mechanism to transfer data from the server to any client. The inventors knew how to use the words “push” and “pull” to describe data transfers and purposely did not limit the Challenged Claims to being implemented via push or pull. Thus, the fact that Brown may use a client-pull mechanism to *initiate* synchronization with server 105 and Dr. Shenoy’s arguments that attempt to mischaracterize Brown in combination with Hesselink as exclusively teaching, suggesting, and rendering obvious embodiments that use pull mechanisms are ultimately irrelevant. There is nothing in the Challenged Patents or their file histories, such as a clear disclaimer or disavowal of claim scope or lexicography, that could justify limiting the claims to cover embodiments in which no client requests are sent to Brown’s server. In fact, the Challenged Patents teach the

opposite: a client ***must*** respond to a server's polling request to allow the server to first determine that the client is in communication with the server. '942 patent, 8:45-59. Even though the Challenged Claims do not expressly recite using such a client request, the Challenged Claims use the word "comprising," which means that Brown in combination with Hesselink disclose and render obvious the Challenged Claims even though they may contemplate certain automated communications between clients and the server that are not expressly recited in the Challenged Claims.

195. Even if the Board were to conclude that one or all of Brown's embodiments, as modified in view of Hesselink, are limited to pull-based HTTP mechanisms as Dr. Shenoy incorrectly asserts, it would have been obvious in view of the Hesselink embodiments discussed above disclosed in connection with paragraphs 104-115 of Hesselink to push data from server 105 to a client. Use of such functionality would have been in the background knowledge of the POSA, as demonstrated by the fact that, e.g., such paragraphs disclose the use of such HTTP messages without even describing what they are. *See* Hesselink, ¶¶ 104-15. The embodiments describe pushing data from server 105 to clients without the need to go into any detail about HTTP or other well-known file transfer methods that could have been used to implement Brown's and Hesselink's systems and methods that implement such functionality. There should be no dispute that such push-based functionality was in the background knowledge of a POSA, given that Dr. Shenoy

himself relies upon evidence that confirms this understanding. *See* IETF HTTP/1.1, §§ 9.5, 9.6.

196. In addition, neither Brown nor Hesselink *requires* use of HTTP/HTTPS, and a POSA would have known how to implement push-based file synchronization using push-based applications. The Challenged Patents refer to “a browser *or other applications*, running on one or more of the user devices 210, 280 [that] can cooperate in two-way communication with server 230 and one or more applications running on server 230.” ’942 patent, 7:51-54. The Challenged Patents provide no detail about how to implement any of these applications, confirming that doing so would have been routine, obvious, well-known, and conventional prior to November 2008. Indeed, there were many well-known, conventional examples of applications used to push data from a server to a user’s device that were in public use prior to November 2008, as discussed above in § VI with respect to the Wong reference.

197. Moreover, Brown expressly states: “From the user’s perspective, the synchronization process is automatic, runs in the background and requires minimal monitoring or intervention.” Brown, 3:40-42. As Dr. Shenoy himself explains, the Challenged Claims are “user-centric[.]” Shenoy Decl., ¶ 22. Thus, all that is required is that the synchronization process be automatic *from the user’s perspective*. Dr. Shenoy’s own statement in paragraph 22 of his declaration

therefore contradicts his later statement in paragraph 199 that the transfer of the modified file from server to client 135 must be automatic from the *server's* perspective. *Id.*, ¶¶ 22, 199.

198. Dr. Shenoy incorrectly states: “While Brown’s file synchronization requires the client to execute prescribed steps, Brown expressly teaches its ‘automatic synchronization’ as being automatic from the *user’s* perspective as long as the client does not necessitate any action from the user in order to initiate the synchronization.” Shenoy Decl., ¶ 200 (emphasis in original). In my opinion, this statement by Dr. Shenoy misrepresents the disclosures in Brown. Brown does not state or even suggest *anywhere* in its disclosures that *any* “action from the user” is necessary “in order to initiate the synchronization.” *Id.* Indeed, Dr. Shenoy fails to identify any such disclosure in Brown in his declaration.

199. I disagree with Dr. Shenoy’s statement that “the automatic sending in the claims of the ’942 Patent is automatic from the *server’s* perspective[.]” *Id.*, ¶ 201 (emphasis in original). To the contrary, claim 4 of the ’561 patent recites: “the first application [running on server 105] is configured to store the modified first electronic file copy to the memory device associated with the first electronic device [server 105] when the third electronic device [client 135] is not in communication with the first electronic device [server 105].” Claim 4 depends indirectly from claim 1, and similar language in claim 4 appears in claim 11, which depends indirectly

from claim 8. In order for claims 4 and 11 to make any sense in the context of their respective independent claims, the independent claims must be read to **not** require synchronization to be automatic “from the server’s perspective,” because the server has absolutely no control whatsoever over whether or when any particular client may go offline or whether or when it may attempt to reestablish communications with the server. In other words, in at least the embodiments claimed in claims 4 and 11, the server **cannot** automatically send any file to an offline client without the offline client first coming back online, reestablishing communication with the server, and requesting the file (whether it be as part of—or as a result of—the reestablishment of communication or otherwise).

b. A POSA Would Have Been Motivated to Look to Hesselink in View of Brown and Would Have Found It Obvious in View of Brown and Hesselink to Automatically Send a Modified File from Brown’s Client 130 to Server 105 and Then to Client 135

200. Dr. Shenoy states: “A POSA would not rely on Hesselink at ¶ 183 as disclosing teaching automatic push from the server.” Shenoy Decl., ¶ 204. I disagree. As discussed above, Hesselink discloses in paragraphs 104-115 embodiments that are strikingly similar to those described in Brown, and Hesselink further discloses “when data is updated, such as when a file is edited, for example, the system may be configured to automatically update all or some designated subset of computers on the network as soon as the update has been performed at one

location.” Hesselink, ¶ 183. A POSA would understand this statement in paragraph 183 of Hesselink to refer to the “system” as a whole, including the embodiments depicted in Figs. 7 and 8 and described in paragraphs 104-115. Dr. Shenoy is therefore incorrect in stating: “Hesselink at the cited portion or anywhere else does not contain any disclosure of the server automatically pushing the modified file.” Shenoy Decl., ¶ 204. To the contrary, as discussed above, the embodiments in paragraphs 104-115 expressly disclose automatically pushing modified files from a source client to a connection server and then from the connection server to a destination client. Hesselink, ¶¶ 104-15. Paragraph 183 only reinforces the idea that it would have been obvious in view of Brown in combination with Hesselink to automatically send an updated file from a source client to a server and from the server to a destination client “as soon as [an] update [on the file] has been performed[.]” *Id.*, ¶ 183.

201. Dr. Shenoy opines: “Hesselink only discusses the pull-based synchronization throughout its disclosure, for example, as illustrated in detail with reference to FIGS. 4 and 8.” Shenoy Decl., ¶ 205. I disagree. Like the embodiments associated with Figs. 7 and 8 of Hesselink discussed above, the embodiment associated with Fig. 4 of Hesselink expressly discloses pushing “data” from a source client through a connection server to a destination client. For example, step 450 described in paragraph 87 of Hesselink involves the sending of data from a source

client (e.g., any one of items 18a, 18b, 18c, 18d, 18e, 18f, 18n-1, or 18n in Fig. 2) through a server (e.g., any of items 14a, 14b, 14c, 14d, 14e, 14f, 14g, 14h, or 14n in Fig. 2) to a destination client (e.g., any one of items 12a, 12b, 12c, etc. in Fig. 2). Hesselink, ¶ 87. Hesselink explains that “full duplex connections” between the “12” clients and the “18” clients may be established, and a POSA would understand or have found it obvious in view of these disclosures that a modified file from a source client would have been sent to a connection server and then forwarded automatically to a destination client. *Id.*

202. Contrary to Dr. Shenoy’s assertions, this embodiment associated with Fig. 4 in Hesselink does **not** use a “pull mechanism.” Shenoy Decl., ¶ 207. Instead, paragraph 87 of Hesselink clearly describes a **push** mechanism whereby data from one or more of the source clients (designated as items with labels beginning with “18” in Fig. 2) push data through a connection server to a destination client (designated as items with labels beginning with “12” in Fig. 2). Hesselink, ¶ 87. After a connection is established, there is no requirement for a client to initiate any request for any update to or from the connection server, contrary to Dr. Shenoy’s assertions—content is merely pushed to the client unsolicited in HTTP responses.

203. Likewise, as discussed, Hesselink’s embodiments depicted in Figs. 7 and 8 involve the use of a push mechanism to send data from a source client, through a server, to a destination client automatically. Therefore, I also disagree with Dr.

Shenoy's assertions that the embodiment associated with Fig. 8 in Hesselink is "focused on a data transfer process based on client pull." Shenoy Decl., ¶ 207.

204. In addition, as discussed above, I disagree with Dr. Shenoy's assertion (which he repeats in paragraph 208 of his declaration) that a POSA would have only known how to use HTTP to pull data from a server, not push data. *Id.*, ¶ 208. As discussed above in § VI a POSA would have understood and found it obvious in view of references like Hepworth and Dr. Shenoy's own cited references regarding HTTP that HTTP could be used to implement data pushes from a server. Moreover, Hesselink does not even require the use of HTTP, and a POSA would have found it obvious to use any available communications protocol to implement data pushes from the "connection servers" discussed in Hesselink to destination clients. Dr. Shenoy himself quotes excerpts from Hesselink that disclose a connection server 14 pushing data from "network-enabled device" 18 to "client computers 12[.]" Shenoy Decl., ¶ 208 (quoting Hesselink, ¶ 78 ("client computers 12 a, 12 b may each comprise programming 32 stored in memory that allow client computers 12 a, 12 b to send instructions to network-enabled devices 18 a, 18 b as requests through connection server 14, ***and receive data from network-enabled devices 18 a, 18 b via responses through connection server 14***")). As discussed above, the fact that a client request may initiate such synchronization processes is irrelevant.

205. Dr. Shenoy discusses yet another example from Hesselink that uses a

server to push a modified file from one client to another client. Shenoy Decl., ¶ 209 (quoting Hesselink, ¶ 141). The embodiment discussed in paragraph 141 of Hesselink is depicted in Fig. 10, which, again, shows multiple clients (72, 74, 76) all connected to a “connection server” (“14”). See Hesselink, Fig. 10, ¶ 141. Paragraph 141 explains that one client (72) may send a file modification to another client (74), which in turn may broadcast the file modification to all other “subscribed user modules.” *Id.*, ¶ 141. Such pushing of file changes to “subscribed” clients, when implemented using the system in Fig. 10, would necessarily make use of, or at least render obvious the use of, the connection server depicted in Fig. 10, thereby necessarily making use of a push-based transfer of content from, e.g., client 74 through a connection server to the destination client, e.g., client 76 receiving subscribed file modifications. *Id.*, Fig. 10. The fact that in some embodiments an initial request from a client must be made to establish an active subscription (*see id.*, ¶ 140) does not change the fact that all subsequent data transfers from one client through the connection server to a destination client are performed using a push-based model from the server to the destination client—in fact, Dr. Shenoy does not appear to dispute this point in connection with the embodiments discussed in paragraphs 140-141 in Hesselink. See Shenoy Decl., ¶ 209. The initial request to subscribe does not change the fact that all subsequent data transfers are automatic, consistent with what is claimed in the Challenged Claims.

c. Hesselink Does Not Teach Away from Brown's Disclosures Related to Synchronizing Using a Server and Two Devices All Associated with the Same User

206. As discussed in my Opening Declaration, Brown expressly discloses two user devices (computers 130 and 135) and a server (105) that are associated with the same user. *See* Balakrishnan Open. Decl., ¶¶ 585-87. Dr. Shenoy does not appear to dispute these opinions anywhere in his declaration. Instead, he asserts: “Hesselink fails to disclose the first application associated with a user automatically transferring a modified first electronic file copy to the third electronic associated with a same user.” Shenoy Decl., ¶ 211. I have been informed by counsel that not every limitation needs to be disclosed in every prior art reference relied upon in an obviousness combination. Based on this principle, it is irrelevant whether Hesselink discloses clients and a server associated with the same user, because I am relying on Brown’s teachings for this limitation, not Hesselink’s.

207. Nevertheless, Hesselink does not teach away from synchronizing using a server and multiple client devices associated with the same user. For example, Hesselink teaches in paragraph 84 that the connection servers (14) shown in Fig. 2 may be operatively coupled to a security server 58 shown in Fig. 3. Hesselink, ¶ 84. In turn, the security server uses an “authentication application” to “see whether or not the authentication data in the request” from a client computer (12) “verifies the *user* of client computer 12” such that the client computer is assigned a connection

server (14), as discussed in paragraph 86 of Hesselink. *Id.*, ¶¶ 85-86. Thus, Hesselink expressly discloses association of a particular user to both client computer 12 and an “assigned” connection server 14. *Id.*, ¶ 86. Subsequently, a connection is established between *the same* connection server 14 that was “assigned” to client computer 12 and a “network-enabled device” 18 “*selected by the user.*” *Id.* Thus, both clients 12 and 18 and the “assigned” server 14 are all expressly described as being associated with *the same* user in Hesselink.

208. Dr. Shenoy is thus incorrect in suggesting that all client computers 12 may be connected to all network-enabled devices 18. Shenoy Decl., ¶ 216 (citing Hesselink, ¶¶ 76, 120-21). This is not what Hesselink teaches. Dr. Shenoy cites paragraph 120 of Hesselink (*see* Shenoy Decl., ¶ 216), which states: “The present invention enables a user to access devices 18a-18n of all the computers *that a user would normally have access to*, from any single location throughout the world where the user has network access, assuming all of the computers also have network access.” Hesselink, ¶ 120. This expressly teaches that the user only has access to computers “that a user would normally have access to,” like a work computer—in other words, computers that are associated with the user. This is consistent with the disclosures in connection with Fig. 10 and paragraph 132 of Hesselink cited by Dr. Shenoy, which merely confirm the customizability of the computers to which a user may have access. Shenoy Decl., ¶ 217 (citing Hesselink, Fig. 10, ¶ 132).

209. Dr. Shenoy is simply wrong in stating that “Hesselink does not make a determination as to whether the third electronic device is associated with the same user by use of the first, second, and third applications recited in the asserted claims, and hence Hesselink cannot automatically send a modified file based on such determination.” Shenoy Decl., ¶ 217. As discussed above, Hesselink, expressly teaches deciding, at the outset, whether the first, second, and third devices (i.e., client 12, connection server 14, and network-enabled device 18) are authorized to communicate with each other and, if so, thereby confirming all three devices must necessarily be associated with the same user.

210. Dr. Shenoy cites paragraph 127 of Hesselink and opines: “Hesselink does not teach a dedicated file transfer and/or synchronization application as recited in the asserted claims.” Shenoy Decl., ¶ 217. Petitioners and I rely upon the applications discussed in Brown for this claim limitation, not Hesselink. Moreover, such applications *are* disclosed in Hesselink in, e.g., the form of software modules 72su and 74su as discussed in paragraph 127 (and other modules and applications discussed elsewhere in Hesselink). By virtue of the fact that these applications are associated with devices that must have been authenticated by the security server, as discussed above, in order to communicate with each other, they are necessarily associated with the same user.

**d. A POSA Would Have Been Motivated to Combine
Brown and Hesselink and Found the Challenged
Claims Obvious in View of Such Combination**

211. As discussed above, I do not believe that the Challenged Claims require either a “push”- or “pull”-based model for file transfers from, e.g., the “first electronic device” to the “third electronic device” in claim 1 of the ’561 patent. The inventors knew how to distinguish between these concepts and nevertheless chose not to specify one or the other (or both) in the Challenged Claims.

212. Dr. Shenoy states: “a POSA would not be motivated to modify Brown’s disclosure to that of a push-based file transfer due to the differences between the pull and push-based methods.” Shenoy Decl., ¶ 219. Even if a push model were required to practice the claims (it is not), I disagree with this statement from Dr. Shenoy. Brown expressly contemplates the concept of pushing changes using server 105’s SFS application to syncing clients: “The server SFS uses this value to quickly find directories with changes that need to be *pushed down to a syncing client*. Directory items with DSI values greater than the value passed in by the polling client SA identify the directories with changes.” Brown, 5:36-40. In addition, for the reasons discussed above, push-based messaging would have been well-known and within the background knowledge of a POSA. Thus, it would have been obvious to a POSA that such push-based messaging could be used in connection with Brown’s embodiments, particularly in view of Hesselink’s teachings, which expressly teach

push-based messaging, as discussed in my Opening Declaration and above.
Balakrishnan Open. Decl., ¶¶ 807-35.

213. Dr. Shenoy quotes Brown’s desire to “minimize the load on the server processor[.]” Shenoy Decl., ¶ 221 (quoting Brown, 3:17-28). This desire would have motivated a POSA to look to a reference like Hesselink for teachings related to how to synchronize files as quickly as possible by immediately and automatically sending modified files received from one client computer associated with a user to another client computer associated with the same user using HTTP responses as taught in Hesselink. Pushing modified files as quickly as possible using, e.g., Hesselink’s “subscription” method discussed above, would minimize the burden on Brown’s server. Thus, I disagree with Dr. Shenoy that “the intended principle of operation and purpose of Brown teaches away from using the server-push mechanism for file synchronization[.]” Shenoy Decl., ¶ 222, and I disagree with his statement that using push-based messaging would “fundamentally change the operation of Brown’s system[.]” *id.*, ¶ 223. Instead, a POSA would have recognized push-based messaging as an obvious alternative that could be used in Brown’s system in view of Hesselink’s teachings, as discussed in my Opening Declaration.
Balakrishnan Open. Decl., ¶¶ 807-35.

214. Dr. Shenoy references Hesselink’s “peer-to-peer embodiment” associated with Fig. 11 of Hesselink and argues that it somehow teaches away from

combining teachings of Brown and Hesselink. Shenoy Decl., ¶¶ 224-32. I disagree. First, I disagree with Dr. Shenoy's statement that the embodiment described in paragraph 183 of Hesselink is solely related to the embodiments associated with Figs. 11-13 in Hesselink. *Id.*, ¶ 225. As discussed above, the embodiment associated with Fig. 10 includes a connection server, and paragraph 183 of Hesselink applies to that embodiment to at least the same extent it applies to the embodiments in Figs. 11-13.

215. Like Sigurdsson, Hesselink describes peer-to-peer and client-server architectures as being well-known, conventional, and obvious variants for implementing the systems and methods of Hesselink. *See, e.g.*, Hesselink, ¶ 126 ("Computers 72 and 74 are connectable across a wide area network by any of the means described above with regard to FIG. 10, for example, via a connection server 14, or directly connected, with or without a VPN."). Thus, a POSA would not have viewed Hesselink as teaching away from Brown's client-server-based systems and methods—as discussed above, Hesselink itself discloses multiple embodiments that are based on a client-server architecture. Thus, it is not solely based on a peer-to-peer architecture.

216. Contrary to Dr. Shenoy's assertions (*see* Shenoy Decl., ¶ 229), the security concerns in paragraph 6 of Hesselink are not relegated solely to peer-to-peer implementations of file sharing. *See* Hesselink, ¶ 6 ("While it is possible to upload

digitized photos to a web page provided by a photo service center, this solution leave some things to be desired too, since many users may be uncomfortable storing photos of their families on some photo service server in some unknown location where it is also often unknown what level of security (if any) such server is provided with.”). Instead, these concerns are addressed by Hesselink’s security server, as discussed above, and the security server is expressly disclosed in Hesselink as being used to set up a client-server architecture for file sharing, as discussed above.

217. Dr. Shenoy states: “while Hesselink mentions including a server within its peer-to-peer system, such server only serves as a pipeline or conduit between two peers.” Shenoy Decl., ¶ 230. But Dr. Shenoy does not describe what, if anything, more is required of the server aside from effectively acting as a pipeline or conduit between two peers. A POSA would understand any transfer of a file from a source client to a server and then from a server to a destination client to necessarily involve some storage at the server, even if only temporarily. And Dr. Shenoy does not argue that anything in Hesselink teaches away from Brown’s express teachings related to the server storing a file from a source client before sending it on to a destination client, as discussed in my Opening Declaration. Balakrishnan Open. Decl., ¶¶ 848-62.

218. I disagree with the Patent Owner’s assertion in its Patent Owner’s Response that paragraphs 6 and 13 in Hesselink describe “specific problems arising

in the client-server system that is adopted in Brown.” POR, 61. To the contrary, problems of security, difficulty in installation, cost, and maintenance (*see* POR, 61) would have arisen in *both* peer-to-peer *and* client-server implementations.

IX. NO SECONDARY CONSIDERATIONS OF NON-OBVIOUSNESS

219. I am not aware of any information (such as commercial success, unexplained results, long felt but unsolved need, industry acclaim, simultaneous invention, copying by others, skepticism by experts in the field, and failure of others) suggesting that the claims addressed herein or in my Opening Declaration are not obvious. Dr. Shenoy did not identify or discuss any secondary considerations of non-obviousness in his declaration. *See* Shenoy Decl., ¶ 13 (“I understand that to date secondary considerations have not been raised in these proceedings.”).

X. ENABLEMENT AND REASONABLE EXPECTATION OF SUCCESS

220. I hereby incorporate by reference § IX of my Opening Declaration as if fully stated herein, which explains, in part, why I believe the prior art references I analyzed provide sufficient detail to enable a POSA to practice the limitations of the Challenged Claims without undue experimentation and why I believe there would have been a reasonable expectation of success in combining the references, as discussed above and in my Opening Declaration.

XI. CONCLUSION

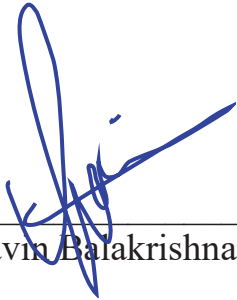
221. I reserve the right to offer additional opinions relevant to the validity of the Challenged Patents and Claims at issue and/or offer testimony in support of this Reply Declaration.

222. In signing this Reply Declaration, I recognize that it will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also recognize that I may be subject to cross-examination in the case. If required, I will appear for cross-examination at the appropriate time regarding the opinions expressed herein.

223. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001.

Reply Declaration of Ravin Balakrishnan, Ph.D. in Support of
Petitions for *Inter Partes* Review of U.S. Patent Nos. 9,143,561 and 10,067,942

Signed:



Ravin Balakrishnan, Ph.D.

Dated:

20 Feb 2024
