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. IPR2023-00430

U.S. Patent No. 11,003,622

Filing Date: March 22, 2019
Issue Date: May 11, 2021

Title: Architecture for Management of Digital Files Across Distributed Network

PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 11,003,622

Table of Contents

Page

I. Introduction....1

II. Mandatory Notices (§ 42.8(**a**)(1)) 1

A. Real Parties-In-Interest (§ 42.8(b)(1))....1

B. Related Matters (§ 42.8(b)(2)) 1

C. Lead and Back-Up Counsel (§ 42.8(b)(3)) 2

D. Service Information (§ 42.8(b)(4))....3

III. Fee Payment (§ 42.103)....4

IV. Requirements Under §§ 42.104 and 42.108 and Considerations Under §§314(**a**)/325(**d**)....4

A. Grounds for Standing 4

B. Identification of Challenge and Statement of Precise Relief Requested 4

C. Considerations Under §§314(a)/325(d)....5

V. Overview of '622 Patent....7

VI. Level of Ordinary Skill in the Art 7

VII. Claim Construction....8

VIII. The Challenged Claims are Unpatentable 8

A. Earliest Effective Filing Date....8

B. Overview of Grounds Including Prior Art Qualification 9

C. Summary of Prior Art....10

B. Ground 1: Obviousness over Sigurdsson with Shappell and Rao....14

Table of Contents

(continued)

Page

C. Ground 2: Obviousness over Brown with Hesselink, Shappell, and Rao. 47

IX. Conclusion 68

List of Exhibits

Exhibit No.	Description of Document
1001	U.S. Patent No. 9,143,561 ("'561" or "'561 patent")
1002	U.S. Patent No. 10,067,942 ("'942" or "'942 patent")
1003	U.S. Patent No. 10,289,607 ("'607" or "'607 patent")
1004	U.S. Patent No. 10,642,787 ("'787" or "'787 patent")
1005	U.S. Patent No. 10,754,823 ("'823" or "'823 patent")
1006	U.S. Patent No. 11,003,622 ("'622" or "'622 patent")
1007	Prosecution History of the '561 Patent
1008	Prosecution History of the '942 Patent
1009	Prosecution History of the '607 Patent
1010	Prosecution History of the '787 Patent
1011	Prosecution History of the '823 Patent
1012	Prosecution History of the '622 Patent
1013	Case No. IPR2022-00782, Paper No. 1, Petition for IPR
1014	Case No. IPR2022-00782, Paper No. 17, Decision Granting Institution of IPR (Redacted – Public Version)
1015	U.S. Provisional Patent Application No. 60/986,896 ("'896 provisional")
1016	U.S. Patent Application Publication No. 2007/0174246 A1 to Johann Tomas Sigurdsson, et al. ("Sigurdsson")
1017	File History of U.S. Patent Application Publication No. 2007/0174246 A1 to Johann Tomas Sigurdsson, et al. ("Sigurdsson File History")
1018	U.S. Patent No. 7,734,690 B2 to George P. Moromisato, et al. ("Moromisato")
1019	U.S. Patent Application Publication No. 2006/0190506 A1 to Rajesh M. Rao, et al. ("Rao")
1020	U.S. Patent Application Publication No. 2005/0091289 A1 to Michael Shappell, et al. ("Shappell")
1021	Microsoft Computer Dictionary (5th ed. 2002) (excerpts)
1022	Jim Boyce, *Microsoft Outlook Version 2002 Inside Out* (2001) ("Boyce") (excerpts)
1023	European Patent Application Publication No. EP 1,621,990 A2 to Kitamaru, et al. ("Kitamaru")
1024	U.S. Patent No. 5,923,325 to Ronald Jason Barber and Edwin Joseph Selker ("Barber")
1025	U.S. Patent No. 7,035,847 B2 to David K. Brown, et al. ("Brown")

List of Exhibits

Exhibit No.	Description of Document
1026	U.S. Patent Application Publication No. 2005/0120082 A1 to Lambertus Hesselink, et al. ("Hesselink")
1027	*Topia Technology, Inc. v. Box, Inc.*, Case No. 6:21-cv-1372-ADA, D.I. 60 (Joint Claim Construction Chart)
1028	Declaration of Dr. Ravin Balakrishnan

I. INTRODUCTION

Petitioners Box, Inc., and Dropbox, Inc., respectfully petition for *Inter Partes* Review ("IPR") of claims 1, 3-5, 7, 11, 12, 14, 16, and 17 ("Challenged Claims") of U.S. Patent No. 11,003,622 ("'622").

II. MANDATORY NOTICES (§ 42.8(a)(1))

A. Real Parties-In-Interest (§ 42.8(b)(1))

Petitioners Box, Inc. ("Box"), and Dropbox, Inc. ("Dropbox"), are the real parties-in-interest to this IPR petition. Petitioners' co-defendants in the pending litigation, Vistra Corp., SailPoint Technologies Holdings, Inc., and Clear Channel Outdoor Holdings, Inc., are not real-parties-in-interest and have had no involvement with this proceeding but are identified here in the interest of over-inclusion.

B. Related Matters (§ 42.8(b)(2))

The '622 is the subject of pending litigation involving Petitioners and Patent Owner Topia Technology, Inc. ("PO") (*Topia Technology, Inc. v. Box, Inc.*, No. 6:21-cv-01372-ADA (W.D. Tex.), *Topia Technology, Inc. v. Dropbox, Inc.*, No. 6:21-cv-01373-ADA (W.D. Tex.)) and involving an unrelated defendant (*Topia Technology, Inc. v. Egnyte, Inc.*, No. 1:21-cv-01821 (D. Del.)).

Petitioners are also filing IPR petitions challenging the asserted claims of five related patents PO also asserts against Petitioners in litigation: U.S. Patent Nos. 9,143,561; 10,067,942; 10,642,787; 10,754,823; and 10,289,607 ("Related Patents"). All of the patents purport to claim priority to the same provisional and

non-provisional applications, and all share substantially similar specifications including substantially the same Detailed Description section.

An IPR petition filed by Unified Patents, LLC against certain claims of U.S. Patent No. 10,067,942 was instituted on November 17, 2022. IPR2022-00782, Decision Granting Institution, Paper 17 (EX1014, "Unified ID"). None of the references presented in that proceeding are the same as, or related to, any of the references presented in this Petition.

PO's complaints asserting infringement of the '622 were served on Box and Dropbox on January 5, 2022. This Petition is timely filed.

C. Lead and Back-Up Counsel (§ 42.8(b)(3))

Petitioners provide the following designation of counsel.

LEAD COUNSEL	BACK-UP COUNSEL
Heidi L. Keefe (Reg. No. 40,673) hkeefe@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Ave., NW Suite 700 Washington, D.C. 20004 Tel: (650) 843-5001 Fax: (650) 849-7400	Reuben H. Chen (Admission pro hac vice to be requested) rchen@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Avenue NW Suite 700 Washington, D.C. 20004 Tel: (202) 728-7070 Fax: (202) 842-7899
	Lowell D. Mead (Admission pro hac vice to be requested) lmead@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Avenue NW

LEAD COUNSEL	BACK-UP COUNSEL
	Suite 700 Washington, D.C. 20004 Tel: (202) 728-7070 Fax: (202) 842-7899
	Samuel K. Whitt (Reg. No. 65,144) swhitt@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Avenue NW Suite 700 Washington, D.C. 20004 Tel: (202) 728-7070 Fax: (202) 842-7899

D. Service Information (§ 42.8(b)(4))

This Petition is being served by Federal Express to the attorneys of record for the '622, John Bird and Andrew Taska of Sughrue Mion, PLLC, 2000 Pennsylvania Ave., N.W., Suite 9000, Washington, D.C. 20006. A courtesy copy of this Petition is also being served via Federal Express on PO's counsel of record in the district court cases, *Topia Technology, Inc. v. Box, Inc.*, Case No. 6:21-cv-01372-ADA (W.D. Tex.), and *Topia Technology, Inc. v. Dropbox, Inc.*, Case No. 6:21-cv-01373-ADA (W.D. Tex.).

Petitioners consent to electronic service at the email addresses provided above for lead and back-up counsel.

III. FEE PAYMENT (§ 42.103)

Petitioners request review of 10 claims. Payment of $41,500 is submitted herewith.

IV. REQUIREMENTS UNDER §§ 42.104 AND 42.108 AND CONSIDERATIONS UNDER §§314(a)/325(d)

A. Grounds for Standing

Petitioners certify that the '622 is available for IPR and that Petitioners are not barred or otherwise estopped.

B. Identification of Challenge and Statement of Precise Relief Requested

Petitioners request IPR institution based on the following Grounds:

Ground	Claims	Basis for Challenge
1	1, 3-5, 7, 11, 12, 14, 16, and 17	Obvious by **Sigurdsson** in view of **Shappell** and **Rao**
2	1, 3-5, 7, 11, 12, 14, 16, and 17	Obvious by **Brown** in view of **Hesselink**, **Shappell**, and **Rao**

Submitted herewith is the Declaration of Dr. Ravin Balakrishnan, a qualified technical expert ("Balakrishnan").[1] (Balakrishnan, ¶¶1-12.)

[1] Given the significant overlap in claimed subject matters, Dr. Balakrishnan's Declaration additionally addresses the Related Patents. This Petition cites only the portions relevant here.

C. Considerations Under §§314(a)/325(d)

§314(a): The *Fintiv* factors do not support denial. The pending litigation is in early stages. No claim construction rulings or significant discovery have occurred. The court recently granted Petitioners' motions to transfer venue, and no case schedule has been entered in the transferee court. Petitioners also intend to move to stay pending any instituted IPR.

Petitioners will not pursue invalidity in litigation based on any instituted ground raised herein.

§325(d): Aside from Sigurdsson, the presented references were not cited during prosecution and are not cumulative of cited references. None of the asserted obviousness combinations is cumulative of any obviousness combination considered during prosecution. (Balakrishnan, ¶¶51, 57-58.)

The citation of Sigurdsson during prosecution does not present any basis to deny institution on discretionary grounds. During the original '622 prosecution, the examiner relied primarily upon Sigurdsson to reject PO's originally proposed claims. (EX1012, 118-140.) In response, PO did not identify any error in the examiner's findings, and instead added metadata-priority limitations to the independent claims to overcome the rejections. (EX1012, 101-109.) With this amendment, PO incorrectly asserted that Sigurdsson "simply describe[s] the use of priorities between documents or other data items" and cited only Sigurdsson

paragraph 59 which discusses a "priority algorithm" that assigns priorities to data items. (EX1012, 108-109.) The examiner then allowed the claims, finding "***[i]n view of applicant's amendment, remarks***" that Sigurdsson did not disclose the added limitation of "automatically transfer, based on the first priority being greater than the second priority, the first metadata to the second client device such that the first metadata is transferred to the second client device prior to the copy of the first file being transferred to the second client device." (EX1015, 60.)

As an initial matter, it appears that PO's characterization of Sigurdsson caused the examiner to not fully appreciate Sigurdsson's relevant teachings. As detailed below regarding element [1c], Sigurdsson describes its server system automatically transferring metadata to a second client device prior to transferring an associated modified file to that device. For example, Sigurdsson's server can be configured to automatically transfer metadata associated with a modified file (within a list of "missing content") to a second client device that was offline when the file was modified, ***prior to*** transferring the modified file to that device. (Sigurdsson, ¶¶69-73.) Neither PO nor the examiner cited these relevant passages during prosecution.

Furthermore, as detailed below regarding element [1d], newly presented references that were not cited or discussed by PO or the examiner during prosecution, namely Shappell and Rao, both teach the added limitations and motivate an obvious modification of Sigurdsson's system. (Balakrishnan, ¶¶160-

194.) A person of ordinary skill in the art ("POSA") would have viewed Shappell and Rao as substantially different from, and non-cumulative of, the references cited during prosecution, expressly teaching and motivating transferring metadata with relatively higher priority than associated files. (Balakrishnan, ¶¶58.)

Finally, **Ground 2** does not rely upon Sigurdsson and presents entirely new art that was not cited during prosecution and is not cumulative. (Balakrishnan, ¶¶612-645.)

V. OVERVIEW OF '622 PATENT

A. Claim Listing

Appendix A lists the Challenged Claims.

B. Specification

The '622 relates to file transfer and synchronization between a user's multiple electronic devices, including "automatic modification-triggered transfer of a file among two or more computer systems associated with a user." ('622, Abstract, 1:22-24, 3:55-4:2, 8:32-40.) *Cf.* Unified ID at 7-9 (summarizing specification).

VI. LEVEL OF ORDINARY SKILL IN THE ART

As discussed below, the claims' earliest effective filing date is November 10, 2008. A POSA 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 with storage, synchronization, and human-computer

interaction technologies. (Balakrishnan, ¶¶15-21.) A person could also have qualified with more formal education and less technical experience, or vice versa. (Balakrishnan, ¶¶17, 21.)

VII. CLAIM CONSTRUCTION

Petitioners do not believe express claim construction is presently necessary. In the pending litigation, PO proposes that no terms need construction; Petitioners propose a construction of transferring a file "responsive to" a user modifying the file that is satisfied by the grounds herein. (EX1027.)

VIII. THE CHALLENGED CLAIMS ARE UNPATENTABLE

A. Earliest Effective Filing Date

The '622 claims priority to 11/9/2007 Provisional Application No. 60/986,896 (EX1015) and a 11/10/2008 non-provisional. Patent claims are **not** presumptively entitled to the benefit of a provisional application filing date; PO bears the burden of proving priority. *Dynamic Drinkware v. Nat'l Graphics*, 800 F.3d 1375, 1380-81 (Fed. Cir. 2015); *PowerOasis v. T-Mobile USA*, 522 F.3d 1299, 1305-06 (Fed. Cir. 2008). PO must demonstrate that the provisional "convey[s] with reasonable clarity to those skilled in the art that, as of the filing date sought, the inventor was in possession of the invention." *PowerOasis*, 522 F.3d at 1306 (emphasis omitted); *see also D Three Enters. v. SunModo*, 890 F.3d 1042, 1047 (Fed. Cir. 2018) (priority requires "disclosure"). As Dr. Balakrishnan explains, the provisional fails to

disclose limitations [1b], [1d]-[1f], [11a]-[11d], [16a], and [16c]-[16e],[2] so the claims' effective priority is no earlier than 11/10/2008. (Balakrishnan, ¶¶65, 92-102.) Petitioners reserve the right to respond if PO attempts to demonstrate disclosure in the provisional.

B. Overview of Grounds Including Prior Art Qualification

Each reference qualifies as prior art under 35 U.S.C. §102:

Reference	**Filed**	**Published/Issued**	**§102 (pre-AIA)**
Sigurdsson	1/25/2006	7/26/2007	§102(a)/(b)/(e)
Rao	2/22/2005	8/24/2006	§102(a)/(b)/(e)
Shappell	10/24/2003	4/28/2005	§102(a)/(b)/(e)
Brown	3/15/2002	12/19/2002	§102(a)/(b)/(e)
Hesselink	11/13/2004	6/2/2005	§102(a)/(b)/(e)

Ground 1 relies primarily on **Sigurdsson**, which discloses the claims' core file synchronization features, as the Examiner found during prosecution (and PO did not substantively dispute), with **Shappell** and **Rao** that further render obvious the limitations relating to metadata and priority assignments.

In litigation, PO has claimed a priority date would allegedly pre-date

[2] See **Appendix A** for bracketed labels.

Sigurdsson. Although PO's claim is not substantiated by corroborating evidence (and PO did not attempt to swear behind Sigurdsson during prosecution), Petitioners present a second ground qualifying entirely under §102(b) even assuming the 11/9/2007 provisional date. **Ground 2** relies on **Brown** with **Hesselink**, disclosing the core synchronization features, and **Shappell** and **Rao**.

C. Summary of Prior Art

1. Sigurdsson

Sigurdsson is strikingly similar to the challenged patents, describing synchronizing files across a user's multiple client devices. (Sigurdsson, ¶¶2-14, 26-27.) Figure 4 shows the relevant system:

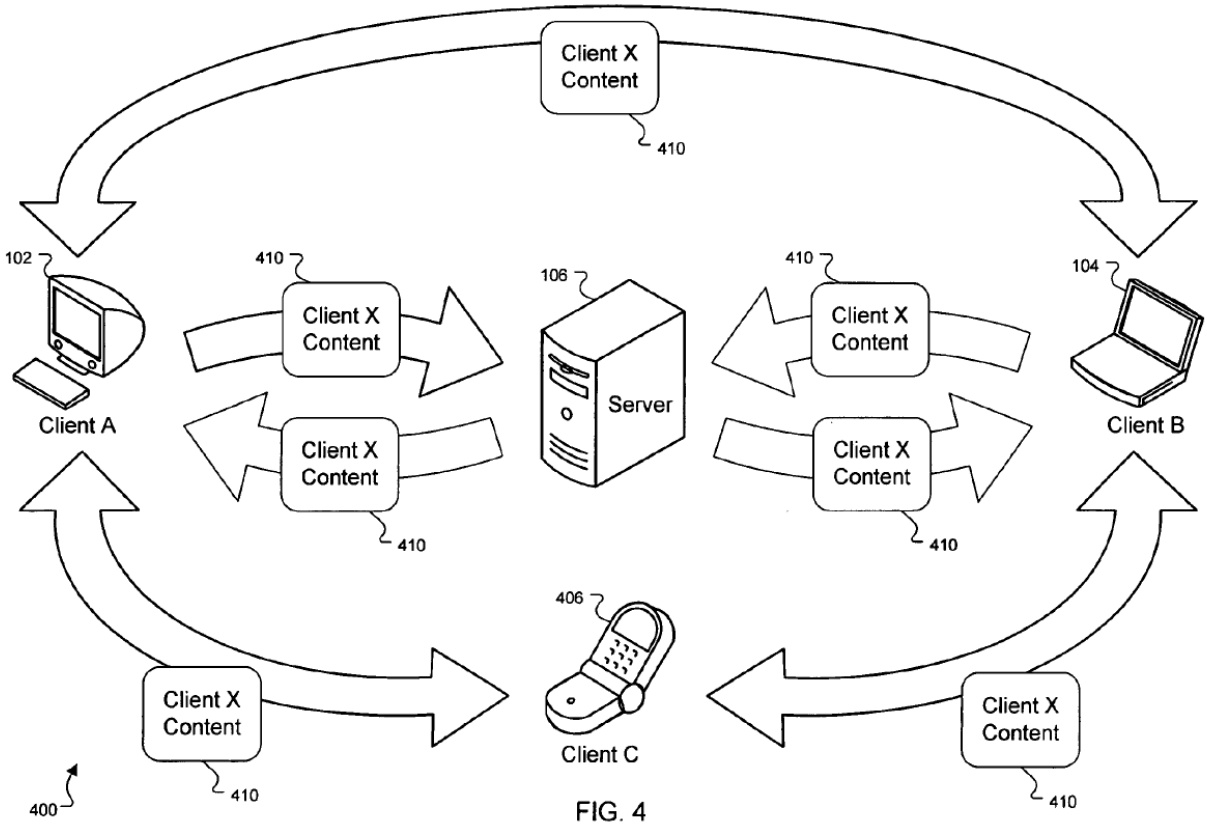


FIG. 4

Clients A, B, and C may be associated with the same user and communicate with a server "through a network," such as the Internet, to transfer new and modified files to the server. (Sigurdsson, ¶¶58, 69, 84, 112-113, 51 ("Although FIG. 4 does not show explicit arrows between the Client C 406 and the Server 106," Client C can "communicate with the Server 106 in a manner similar to the method used by the Clients A or B," including method 200 of Figures 1-2).) "Server 106" in Figure 4 "can provide a temporary storage location, or 'drop box', for a user's files, documents, web pages, or the contents of these documents." (Sigurdsson, ¶27.) Content sent from clients to Server 106 may include copies of files and metadata. (Sigurdsson, ¶54.) Using a "priority algorithm," different items may be prioritized differently for synchronization, based on factors such as "size, type, or age." (Sigurdsson, ¶58.) As a result, metadata may be sent to Server 106 prior to associated files, such as when the files are large. (Sigurdsson, ¶72.) Server 106 can also send a list of "missing" content, including the metadata, to a client that was offline when the missing content was created or modified. (Sigurdsson, ¶73.)

2. Rao

Rao teaches synchronizing files between a client and a server in two phases: first synchronizing metadata associated with files and then synchronizing the raw file data. (Rao, ¶63.) Rao explains that transmitting metadata first, before raw data, is useful because it may enable certain functionality before the raw data is present,

and the "raw data may be extraordinarily large" and "may require a long time to transfer…." (Rao, ¶¶43-44, 46, 48.) For example, metadata may be used to display information about a file that is available on the server not on the client. (Rao, ¶43.)

3. Shappell

Shappell teaches "shared spaces" where files are shared and synchronized among group members. (Shappell, ¶76.) Shappell discloses that a group member computer may transmit file metadata first, when a file is modified, to facilitate user-interface display regarding the availability of the files before the files are transmitted. (Shappell, ¶¶78, 79, 76.) Shappell also discloses an "automatic replication" option that, when enabled, causes automatic replacement of old versions of shared files with new versions when the files are modified. (Shappell, ¶81.)

4. Brown

Brown discloses a client/server system where a user's client devices (e.g., desktop **130** and laptop **135**) synchronize through a server (**105**). (Brown, 2:65-67 ("Client machines synchronize with each other by synchronizing to a common server account.").)

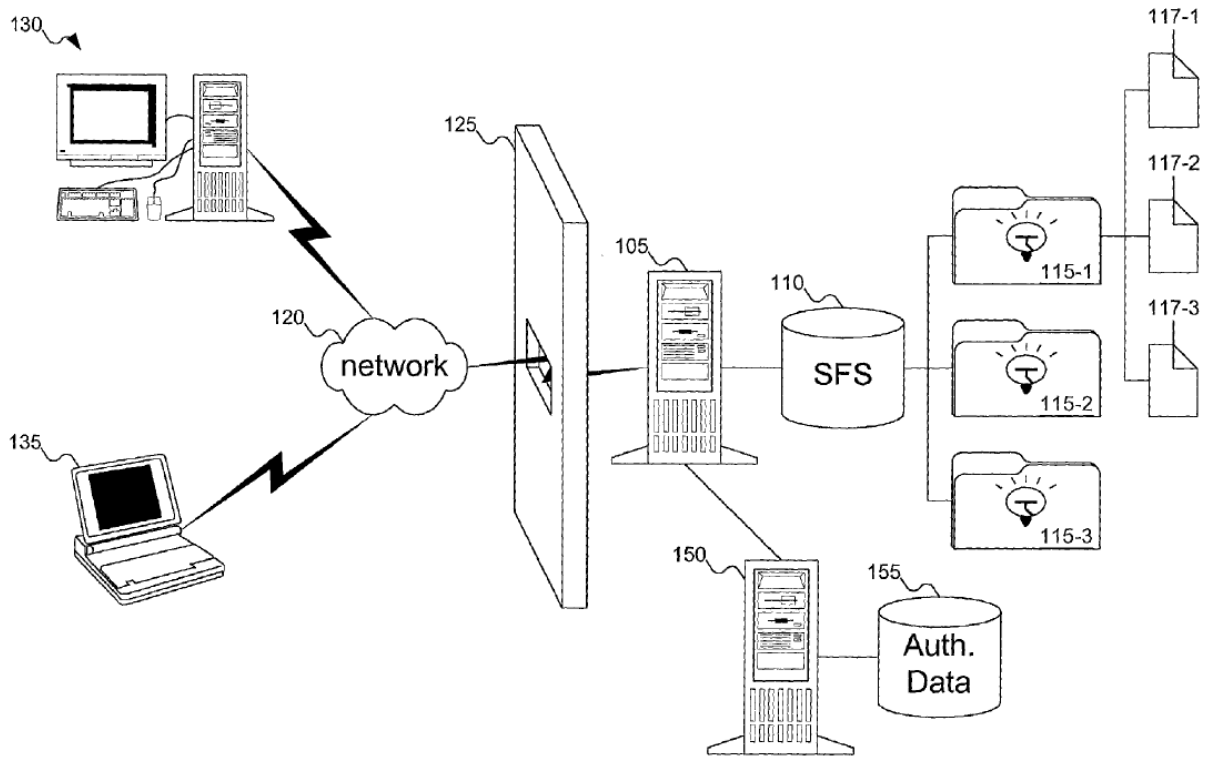


FIG. 1

(Brown, Fig. 1.) "From the user's perspective, the synchronization process is automatic, runs in the background and requires minimal monitoring or intervention." (Brown, 3:40-42.) Brown's synchronization process includes two main steps: first, metadata is exchanged between clients and the server, and second, new or recently updated files are exchanged. (Brown, 5:4-24.)

5. Hesselink

Like Brown, Hesselink discloses a synchronization system involving multiple user computers, such as home, work, and laptop computers. (Hesselink, ¶210.) The system allows a user to be "ensured of having a completely up to date master copy

of each shared data file" on each user device. (Hesselink, ¶192.) Hesselink teaches automatically synchronizing a file "as soon as the update has been performed" at one device, so all connected devices "may be automatically updated concurrently." (Hesselink, ¶¶183, 189, 197.) Hesselink's system "automatically and repeatedly" checks whether synchronizing devices are in communication prior to initiating synchronization. (Hesselink, ¶¶182, 189.)

B. Ground 1: Obviousness over Sigurdsson with Shappell and Rao.

1. Claim 1.

(a) [1pre]

Sigurdsson discloses a "**system**" comprising clients A-C and server 106 as shown in Figure 4:

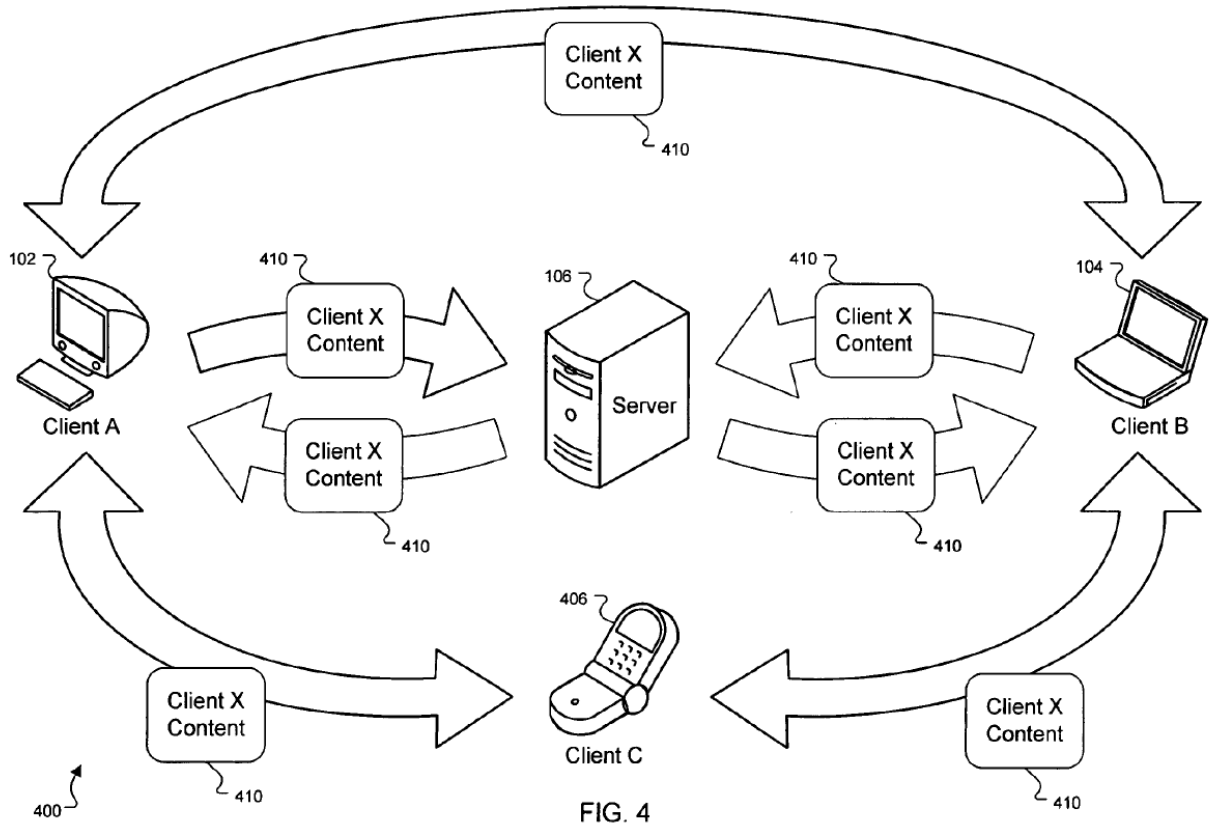


FIG. 4

(Sigurdsson, Fig. 4, ¶¶45-51; Balakrishnan, ¶¶146-149.)

(b) [1a]

Server 106 comprises a "**server system comprising one or more processors programmed with computer program instructions that, when executed, cause the server system to**" perform operations. (Balakrishnan, ¶¶150-152.) Server 106 may include "System 900," including "Processor 910" programmed with executable program instructions. (Sigurdsson, ¶¶104-105, 109 (System 900 "implemented in a computer program . . . , for execution by a programmable processor"); Balakrishnan, ¶151.)

(c) [1b]

Sigurdsson discloses and renders obvious [1b]. (Balakrishnan, ¶¶153-159.) Server 106 "**receive[s], over a network, a copy of a first file from a first client device associated with a user**," such as "Client A" ("**first client device**"). (Sigurdsson, ¶¶8, 27-29, 52, 55, 58, 84, 51 (Figure 4 components can share content using method 200 of Figure 2 implementing Figure 1 client/server system); Balakrishnan, ¶¶153-159.) Each client device is "**associated with a user**," such as the user's "home computer" (Client A), "work computer" (Client B), and mobile device (Client C). (Sigurdsson, ¶¶27-29, 55, 51, Figs. 1-2, 4; Balakrishnan, ¶159.)

Sigurdsson's client devices send content including "**file[s]**" to Server 106 "**over a network**." (Sigurdsson, ¶¶8, 28 (content may be "file"), 69, 112-13, 51, Fig. 4; Balakrishnan, ¶¶153-154.) For example, "Client A Content" such as a "maintenance manual" (exemplary "**first file**") is stored on Client A ("**stored on the first client device**"). (Sigurdsson, ¶58; Balakrishnan, ¶159.)

Server 106 "**automatically receives**" a "**copy of the first file from the first client device responsive to the user modifying a content of the first file**." (Sigurdsson, ¶¶26-28 (when user saves modifications to file, "Client A Content 108, which can include ***a copy of the user's file***, is ***sent automatically*** to the Server 106."), 52 ("edit and save"), 84 ("Client A 102 posts Client A Content 108 to the Server 106 ***when content is changed*** on the Client A 102" and "posting can

occur as soon as an event is generated by an action, such as saving a file"); Balakrishnan, ¶¶156-157.) The "**copy of the first file [is] an updated version of the first file that is generated from the user modifying a content of the first file**." (Sigurdsson, ¶¶26-28, 58 (exemplary "maintenance manual" and "update[d]" "office memo" synchronized via Server 106), 84; Balakrishnan, ¶¶157, 159.) A POSA would have understood that performing a file-save operation on an edited file would result in modifying a content of a stored file (i.e., "**modifying a content of the first file *stored on the first client device***"). (Balakrishnan, ¶¶156-158; EX1021 at 220, 463.)

Although Figure 4 of Sigurdsson depicts both client/server and peer-to-peer connections, a POSA would have appreciated Sigurdsson's teachings of the obvious advantages of a client/server arrangement. (Balakrishnan, ¶¶154-155.) The server can stay online while user computers periodically go offline, so that Server 106 beneficially provides a "drop-box" for "a user's files" to "***facilitate sharing this information between the user's computers by storing the shared content if the target computer is offline***." (Sigurdsson, ¶¶27-29, 61 (server tracks clients' online/offline status); Balakrishnan, ¶155.) A server may also have more available bandwidth for transferring data. (Sigurdsson, ¶79.) Sigurdsson makes clear the peer-to-peer and client/server pathways between the user's multiple client devices are interchangeable, rendering obvious synchronization through the server. (*Id.*, ¶51

("Although FIG. 4 does not show explicit arrows between the Client C 406 and the Server 106," Client C can "communicate with the Server 106 in a manner similar to the method used by the Clients A or B," including method 200 of Figures 1-2); Balakrishnan, ¶155.) For example, each client may include an "HTTPS Client/server" that may interchangeably be used for synchronization with Server 106 (Sigurdsson, ¶54) ***or*** with other clients (*id.*, ¶47). (Balakrishnan, ¶155.) Even when clients can communicate peer-to-peer, they can still also use Server 106 for content sharing. (Sigurdsson, ¶¶51, 79 ("***Server 106 can still serve as a drop-box for temporarily (or permanently) storing Client X Content 410 originating from any the Clients 102, 104 or 406***."); *Power Integrations v. Fairchild Semiconductor*, 843 F.3d 1315, 1337 (Fed. Cir. 2016) (prior art "that sometimes, but not always, embodies a claimed method nonetheless teaches that aspect of the invention."); Balakrishnan, ¶155.)

(d) [1c]

Sigurdsson discloses [1c]. (Balakrishnan, ¶¶336-339.) Sigurdsson discloses that its Server stores the modified **first file** to a memory device in the Server. (Sigurdsson, ¶¶48 (discussing "Storage 330" in Server 106 containing the content received from clients), 74 ("***the content may be stored at the server***"); Balakrishnan, ¶337.) Server 106 also serves as a "***temporary storage location*** that can facilitate synchronization between the user's clients." (*Id.* ¶¶83-85 ("The Client A Content

108 is received by and ***stored in the Server 106***."); Balakrishnan, ¶337.) Server 106 further includes "a Space Quota 342 and a Throughput Threshold 344" to control data storage. (*Id.* ¶48; Balakrishnan, ¶¶337-338.)

Sigurdsson also discloses "System 900" in Figure 9 and explains that "***the System 900 may be included in*** either or all of the Client A 102, the Client B 104, and ***the Server 106***." (Sigurdsson, ¶104; Balakrishnan, ¶339.) Sigurdsson further contemplates the use of standard memory/storage devices. (*Id.*, ¶110; Balakrishnan, ¶339.)

(e) [1d]

Sigurdsson with Shappell and Rao renders obvious [1d]. (Balakrishnan, ¶¶160-194.) Sigurdsson discloses **first metadata associated with the first file generated from the user modifying the content of the first file**. When the user (e.g., at Client A) modifies a file (e.g., a resume), "[m]etadata describing the content, such as the time the resume was saved, can also be transmitted to the Server 106" along with the file. (Sigurdsson, ¶¶32-34, 5 ("automatically extracting content from the first file in response to the event using the first client and ***generating metadata to associate with the content***"), 65 ("metadata that describes . . . when the data was last changed"), 72-73 ("list of the content that was generated" while Client C was

offline, such as "metadata describing the file and the time it was accessed");[3] Balakrishnan, ¶163.)

Server 106 then **receives over a network, from the first client device** (e.g., Client A), that **first metadata**, along with the updated first file (per claim 1[b]). (Sigurdsson, ¶¶6 ("***transmitting the content and the metadata to the server***."), 35 ("Server 106 updates database tables with the Client A Content 108 ***and any corresponding metadata describing the Content 108 it received from the Client A 102***."); 54 ("In step 506, the content originating on the client is posted to the

[3] Although this exemplary metadata is disclosed in the context of a "peer-to-peer content flow" (Sigurdsson, ¶65), it would have been obvious to incorporate such metadata into Sigurdsson's client-server flow, such as the paragraphs 72-73 example where Client A transfers metadata associated with a large file (e.g., maintenance manual) to the server. *Boston Sci. Scimed v. Cordis*, 554 F.3d 982, 991 (Fed. Cir. 2009) ("Combining two embodiments disclosed adjacent to each other in a prior art patent does not require a leap of inventiveness."); Sigurdsson, ¶¶51, 65, 72; Balakrishnan, ¶163 n.10. For example, a POSA would have been motivated to include metadata about when a large maintenance manual ("**first file**") was last modified so that users know if they have the most up-to-date maintenance instructions. (Balakrishnan, ¶163 n.10.)

server. The content can include ***a copy of the document*** . . . that was saved by the user as well as ***metadata that describes information, such as the type of data, its source, when it was saved, and when it was posted to the server***"), 72-73 (list containing generated metadata "***transmitted to the Server***"); Balakrishnan, ¶164.)

Claim [1d] further recites "**the first metadata being assigned a *priority* that is greater than a second *priority* assigned to the copy of the first file**," and claims [1e] and 3 recite further "priority"-related limitations. (Balakrishnan, ¶165.) For convenience, the Petition will address the interrelated "priority"-related limitations of claims [1d], [1e], and 3 together.

For context in interpreting these priority-related limitations, the '622 specification states only: "An embodiment's directory update system updates at a higher priority than the documents to be synchronized." ('622, 9:12-13; EX1015 at 17; Balakrishnan, ¶166.) No affirmative assignment of priority values is described. Nor does the specification explain which component, if any, controls the "higher priority." Claim [1d] likewise does not recite that "the first metadata being assigned a first priority" must be assigned its priority prior to the server system receiving that "first metadata." On the contrary, claim 3 provides that the server may be the only entity that assigns "the" first priority to the first metadata, which could naturally occur **after** the server receives that metadata. (*See* claim 3; Balakrishnan, ¶166.) Nevertheless, to the extent [1d] requires assigning the "first priority" to the first

metadata **before** the server receives that metadata, the prior art discloses and renders that obvious, as discussed below. (*Id*.)

Under any reasonable interpretation consistent with the specification, Sigurdsson in view of **Shappell** and **Rao** disclose and render obvious claims [1d], [1e], and 3 together providing **the server system to automatically transfer the first metadata (received from the first client device) over a network to a second client device prior to transferring the first file, based on a first, greater priority assigned to metadata by a priority assignment configuration on the first client device or server system that assigns a first, greater priority to metadata and a second, lower priority to files**. (Balakrishnan, ¶¶167-194.)

Under a first interpretation, if these limitations are interpreted to encompass a server system configured to receive metadata from a first device, and then automatically (not user-manually) transfer that metadata to a second client device before transmitting associated files, **Sigurdsson** discloses and renders them obvious (in addition to rendering them obvious with Shappell and Rao). (*Id.*, ¶168.)

Sigurdsson discloses and renders obvious that Server 106 receives from the **first client device** a file's generated metadata ***before*** receiving the file, and then the server automatically transfers that metadata to a **second client device *before*** transmitting the file. (*Id.*, ¶169.) A "priority algorithm" (discussed below), optionally implemented on the clients and/or Server 106, prioritizes the order in

which files are transferred. (Sigurdsson, ¶¶58-60, 72; Balakrishnan, ¶170.) Higher-priority (e.g., smaller and/or newer) files may be transferred before lower-priority (e.g., larger and/or older) files. (*See id*.) As a result of the priority algorithm, Server 106 can first receive from the first client device the ***metadata*** associated with a low-priority ***file*** that Server 106 has not received: "The content of this ***file may not be transmitted to the Server*** 106 ***because of the execution of a priority algorithm*** (as discussed earlier). ***Instead, metadata describing the file and the time it was accessed may be transmitted to the Server 106***." (Sigurdsson, ¶72; Balakrishnan, ¶170.) And although it does not appear to be a requirement of the claim, this "metadata describing the file and the time it was accessed" obviously includes metadata generated from the user modifying the file content (which would change the file's description and last-accessed time), matching the "metadata describing the content, such as the time the [file] was saved" generated when a user modifies a file. (Sigurdsson, ¶34; Balakrishnan, ¶170.)

That metadata received by Server 106 is then transmitted by Server 106 to a **second client device** before Server 106 transfers the associated file. (Balakrishnan, ¶172.) Specifically, that received "***metadata*** describing the file and the time it was accessed . . . ***may be included in a list tracking content*** that was generated" during a period of time when another client (e.g., Client C) was offline. (Sigurdsson, ¶¶69-73; Balakrishnan, ¶172.) Once that client (e.g., Client C) comes back online, it

requests and receives from Server 106 that "list of missing content" including the metadata. (Sigurdsson, Fig. 6 (steps 612, 614, 616), ¶¶69-73; Balakrishnan, ¶172.) That client uses the list to obtain the missing files, which Server 106 may transfer. (Sigurdsson, ¶¶74, 79-80, 96; Balakrishnan, ¶172.)

Accordingly, the client device's request for a list of missing content **configures** Server 106 **to prioritize transmitting the first metadata received from the first client device to the second client device before transmitting the file**. (Balakrishnan, ¶173.)

Under a second interpretation, if the priority-related limitations of claims [1d], [1e], and 3 are interpreted to require affirmative, categorical assignment of greater/lesser priorities to metadata and files, they are additionally rendered obvious by **Sigurdsson** with **Shappell**, which teaches prioritizing transfer of metadata for a modified file before transferring the file itself, and **Rao**, which teaches categorical assignment of first and second "phases" to transfer metadata before files between clients and server. (*Id.*, ¶174.)

As noted, Sigurdsson's overall "system **400**" (including Server 106 and/or client devices) optionally assigns "priority" to items using a "priority algorithm," to address the concern that limited available network bandwidth may cause "content sharing [to] fall behind demand." (Sigurdsson, ¶58; Balakrishnan, ¶175.) For example, "[a]lthough the Server 106 can receive content for all three data items in

the order that the user accessed them, ***the Server 106 can also deal with the data items by assigning it priorities***. ***The Server 106 can handle the higher priority data items first, and save the lower priority data items for later***." (*Id*., ¶¶58-59 (each client may also use the priority algorithm; priority scores may be based on attributes such as "size, type or age"); Balakrishnan, ¶175.)

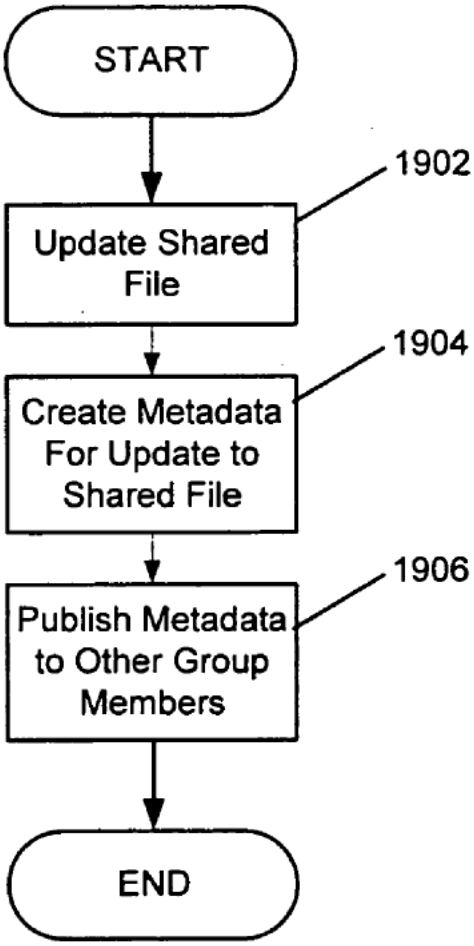
Sigurdsson in view of Shappell and Rao renders obvious a predictable variation where the priority algorithm (**priority assignment configuration**) in the system, configured on the first client and/or Server 106, categorically assigns greater priority to metadata and lower priority to files, thereby causing metadata to transfer prior to the transfer of the associated files based on the greater priority, to advantageously inform clients about updated files. (Balakrishnan, ¶176.)

Sigurdsson itself motivates this modification. (Balakrishnan, ¶177.) The purpose of Sigurdsson's priority algorithm is to prioritize "data items" based on, for example, "size, type or age," in view of potential limited bandwidth, as "content sharing can fall behind demand." (Sigurdsson, ¶58; Balakrishnan, ¶177.) Sigurdsson teaches assigning a greater priority to ***smaller*** and ***newer*** data items—precisely the attributes a POSA would recognize as applicable to ***metadata*** (typically containing much less data than the file itself) for an ***updated*** file—as well as the ***type*** of "data item," which obviously would differentiate metadata from file contents. (Sigurdsson, ¶¶58-59; Balakrishnan, ¶177.) A POSA would also appreciate the

natural harmony between the priority algorithm and Server 106 prioritizing the "list of missing content" (containing metadata) to inform clients about new content at Server 106, and prioritizing high-priority data items. (Sigurdsson, ¶58; Balakrishnan, ¶177.) From these teachings, a POSA would have found it obvious to configure the priority algorithm to assign greater priority to metadata, and lower priority to the associated files, advantageously and predictably causing transfer of metadata before associated files. (Sigurdsson, ¶¶60, 79; Balakrishnan, ¶177.) *Boston Sci.*, 554 F.3d at 991 (obvious to combine two adjacent prior art embodiments); *B/E Aerospace v. C&D Zodiac*, 962 F.3d 1373, 1379-80 (Fed. Cir. 2020) (affirming obviousness of "predictable" variation).

Shappell and **Rao** both further disclose and motivate this obvious modification. (Balakrishnan, ¶178.)

Like Sigurdsson, **Shappell** teaches synchronizing files across multiple computers using metadata. (Shappell, ¶9; Balakrishnan, ¶178.) Shappell discloses "shared spaces" in which files may be shared among group members' computers. (Shappell, ¶¶31, 42, 75-81; Balakrishnan, ¶178.) "When a ***modification of a file***" occurs, "***metadata is transmitted to all members***" of the group. (Shappell, ¶¶81, 76, Fig. 19 (steps 1902-1906); Balakrishnan, ¶178.)



(Shappell, Fig. 19.)

Disseminating the metadata first (with a higher priority) serves an important function: "[t]o ***inform a user*** of a group shared space file that is not present locally" by displaying an indicator at each computer "***to denote a 'missing' file that is available on another system***." (Shappell, ¶¶79, 76 (the "metadata are sufficient to render an icon of the shared file and may include date, time, thumb-nail information, name size, and optionally the source of the information"); Balakrishnan, ¶178.) In one illustration, the indicator is a "ghosted or semi-transparent" icon. (Shappell,

¶¶79, 81; Balakrishnan, ¶178.)

If a version of the modified file already exists locally, the receipt of metadata causes the computer to either "remove the local copy of the out-of-date file," which "will change the visual representation of the file to a 'missing' file," or alternatively obtain "the updated version of the file" if "automatic replication" is enabled. (Shappell, ¶81, Fig. 20 (steps 2002, 2004, 2010, 2006); *id.*, ¶¶62, 77-78; Balakrishnan, ¶178.)

Rao provides naturally complementary teachings to categorically prioritize synchronizing metadata before files between server and client, to advantageously enable functions at the client. (Balakrishnan, ¶180.) Like Sigurdsson, Rao describes file synchronization between a client and a server over a network. (Rao, ¶¶26, 37; Balakrishnan, ¶180.) Rao discloses synchronizing between server and client in ***two phases***: first the metadata, and then the "raw data" of the files themselves. (Rao, ¶63; Balakrishnan, ¶180.) Much like Shappell, Rao teaches that prioritizing metadata enables functions using the metadata beneficial to "the user of the client" receiving the metadata, such as to "display" information about files. (Rao, ¶¶63, 43; Balakrishnan, ¶180.)

Motivation to Combine Sigurdsson with Shappell and Rao

It would have been obvious to combine Sigurdsson with Shappell's and Rao's metadata-priority teachings, predictably resulting in Sigurdsson's system where a

priority algorithm at the first client and at Server 106 assigns a first, greater priority to metadata and lesser priority to files, resulting in the metadata transferring first (based on the first priority) before transferring associated files. (Balakrishnan, ¶181.)

Sigurdsson, Shappell, and Rao are analogous references in the same field as the '622, directed to synchronizing files and associated metadata. (Balakrishnan, ¶182.) The combination would have straightforwardly combined prior art elements according to known techniques to predictably improve Sigurdsson's system in the same way the relevant features enhance Shappell's and Rao's disclosed systems. (Balakrishnan, ¶183.)

A POSA also would have been motivated to consider both Shappell and Rao given their originations with leading file-management vendor Microsoft, as Sigurdsson likewise contemplates standard "home" and "work" computers with Microsoft software. (Sigurdsson, ¶¶27, 42, 45, 58, 60; Balakrishnan, ¶184.)

Furthermore, a POSA would have been motivated to consider additional ways to address Sigurdsson's stated bandwidth concerns. (Sigurdsson, ¶¶51, 58 (with limited bandwidth, "content sharing can fall behind demand"), 60; Balakrishnan, ¶185.) A POSA would have recognized that Sigurdsson's system gives rise to a potential gap in synchronicity between versions of a synchronizing "very large document" over a slow network connection, without notifying the user. (Sigurdsson,

¶58; Balakrishnan, ¶185.) Indeed, as Rao cautions, "synchronizing" a very large set of files "may take many minutes or even several hours, depending on the throughput of the communication path . . . among other things." (Rao, ¶48; Balakrishnan, ¶185.) As an illustration, the user of one client device (e.g., work computer) in Sigurdsson may work on modifying a large file. (Balakrishnan, ¶185.) When the user subsequently accesses another client device (for example, home computer) to continue work on that file, the user would benefit from being able to quickly and visually determine whether or not the version of the file on the home computer is current, before opening the file to work on it. (Balakrishnan, ¶185.) In view of this potential deficiency in Sigurdsson, a POSA would have been motivated to consider Shappell and Rao for a solution that prioritizes transmitting metadata generated by modifying the file to the user's other client devices (including through Server 106, using its priority algorithm, to account for other client devices that may be currently offline), advantageously leveraging the metadata for "any purpose" such as to "***inform a user***" that an updated file is available. (Shappell, ¶79; Rao, ¶¶63, 43; Balakrishnan, ¶185.)

More generally, the benefits of prioritizing transferring metadata before associated content would have been background knowledge of a POSA from well-known commercially-available software. (Balakrishnan, ¶186.) For example, Microsoft Outlook and Outlook Express downloaded headers for email and

newsgroup messages, comprising metadata associated with the messages (e.g., subject, date, etc.), before downloading the messages themselves. (Boyce, 366-370, 3017-308; Balakrishnan, ¶¶186-187.) Boyce notes an "advantage" of this functionality, "the ***ability to work with the messages headers of waiting messages without having to download the messages themselves***." (Boyce, 366; Balakrishnan, ¶186.) For example, prioritizing metadata/headers "is extremely useful when you're pressed for time but have a message with a large attachment waiting on the server," enabling prioritizing messages to receive using limited time and resources. (Boyce, 366, 196 (optionally download headers for messages larger than a specified size); Balakrishnan, ¶186.) A POSA would have appreciated that these advantages of prioritizing metadata for messages—as well-known ways of sharing files/content—would have similarly applied to the Sigurdsson, Shappell, and Rao file sharing/synchronization systems. (Balakrishnan, ¶188.)

A POSA would have reasonably expected success in implementing this modified Sigurdsson system, as Sigurdsson already discloses metadata and file transfer using standard, familiar computer, OS, and network components. (Balakrishnan, ¶189.) A POSA would not have been dissuaded by Shappell's primary embodiment being peer-to-peer rather than client-server, as Shappell's relevant features are independent of these types of network architectures. (Balakrishnan, ¶189.) While Shappell's exemplary embodiment is "server-less"

(e.g., Shappell, Abstract, ¶7), Shappell does not specifically disparage client/server implementations, proclaims that the "type of network . . . is not critical" and is only "preferably" peer-to-peer, and suggests also transferring "indirectly" between machines such as a client/server architecture would provide. (Shappell, ¶40; Balakrishnan, ¶189.) Similarly, a POSA would not have been dissuaded by Shappell's statement that devices "typically" would have different users (Shappell, ¶40), given the compelling applicability of Shappell's relevant teachings to improve Sigurdsson's system. (Balakrishnan, ¶¶189-194.)

(f) [1e]

Sigurdsson with Shappell and Rao disclose and render obvious [1e] as explained for [1d].

(g) [1f]

Sigurdsson with Shappell discloses and renders obvious obvious [1f]. (Balakrishnan, ¶¶340-348.) As explained for [1b], Sigurdsson's clients communicate with Server 106 **over a network**. (Balakrishnan, ¶341.) Sigurdsson describes 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, ¶343.) The transfer of "Client X Content 410" can be sent automatically to a client that was temporarily offline when it comes

back online. (Sigurdsson, ¶69 ("Step 612 can occur ***after the third client comes online, having been offline during the time that the content originated on the first client and was propagated to the second client and to the server***. In step 612, the third client requests a list of missing content from the server. For example, ***the Client C 406 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***."). "In step 614, the server determines the content missing from the third client." (Sigurdsson, ¶70; Balakrishnan, ¶343.) "In step 616, ***the server passes the list of missing content to the third client*** . . . , which had been offline." (Sigurdsson, ¶73; Balakrishnan, ¶343.)

A POSA would have found [1f] obvious regarding Clients A and C as described in Sigurdsson Figure 6 and paragraphs 72-74 and 79-80. (Sigurdsson, ¶¶72-74, 79-80, Fig. 6; Balakrishnan, ¶¶344.) Namely, Client A would send the "**first file**" to the server, which, in turn, would determine that it is in communication with Client C (after Client C has been offline and resumed communications, for example). Thereafter, the server would send Client C the "list identifying the missing content" referenced in paragraph 73 and, as described in paragraph 79-80, would send missing content (i.e., the "**first file**") to Client C. (Sigurdsson, ¶¶73, 79-80; Balakrishnan, ¶344.)

Transferring the first file to Client C would occur **automatically**, with no user intervention. (Balakrishnan, ¶345.) Moreover, given the fungibility of Sigurdsson's clients, it would have been obvious to receive a file from Client A at the Server and then automatically transfer the file to Client C in view of the embodiment described in connection with Figure 5, which describes a similar data flow between Clients A and B. (Sigurdsson ¶¶51-55 ("In step 508, the server stores the content received from the originating client, and the content is propagated to the second client.").) (Balakrishnan, ¶346.)

Sigurdsson's discussion of synchronization would have motivated a POSA to consider the problem of a user having copies of the same file on their home and work computers and synchronizing both computers to make sure they are updated with the most recent updates. (Balakrishnan, ¶347.) For example, Sigurdsson discusses a client's work and home computers as two exemplary "clients," and describes example use cases such as a user editing a Word document synchronized between those computers. (Sigurdsson, ¶¶26-28 (also describing Server 106 as optional "drop box" "conduit" between client devices); Balakrishnan, ¶347.)

A POSA would have found it obvious that after an updated version of the document is transmitted to the user's other device, the user may continue editing it, but Sigurdsson does not detail what would happen to an outdated prior version of that file at the receiving device in that circumstance. (Balakrishnan, ¶348.) A POSA

would have been motivated to look for solutions to address outdated versions of files, and would have found it obvious that one possibility would be for the updated modified file to **replace** the outdated older version, for reasons including the obvious benefit of preventing copies of outdated versions from endlessly accumulating on the user's devices. (Balakrishnan, ¶348.)

Furthermore, **Shappell** teaches and motivates this feature with its "automatic replication" option. Shappell's "automatic replication" option renders obvious implementing Sigurdsson's system so that the receipt of the copy of the first file at the second client device will **replace an older version of the first file stored on the second client device**. (Balakrishnan, ¶224.) Shappell teaches a user-selectable "File Replication" setting where the user can 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; Balakrishnan, ¶224.) With "automatic replication," an "out-of-date file" is "remove[d]" from the receiving computer and replaced with an "updated version." (Shappell, ¶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; Balakrishnan, ¶224.)

It would have been obvious to implement Shappell's "automatic replication" feature as an option for each Sigurdsson client device, including Client C (the "**second client device**"), so that Server 106 would **automatically** transfer to Client C the copy of a file modified at Client A (the "**first client device**"), to ensure that the user at Client C would have the most recently updated file. (Balakrishnan, ¶225.) 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; Balakrishnan, ¶225.) The motivations to combine Sigurdsson with Shappell and Rao for [1d] apply here. (Balakrishnan, ¶¶160-194.)

2. Claim 3.

As explained for [1d], Sigurdsson with Shappell and Rao discloses and renders obvious claim 3.

3. Claim 4.

Shappell renders obvious claim 4's limitations. As discussed for [1d], Shappell teaches that when a file is modified at one computer, metadata is

transmitted and received by other computers, causing those computers to "***change the visual representation of the file to a 'missing' file … that is available on another system***" and is "ghosted" or semi-transparent, when the computer is not set to an optional automatic-replication setting. (Shappell, ¶¶79, 81, Figs. 19-20 (steps 1902, 1904, 1906); Balakrishnan, ¶¶205, 224.) Additionally, the icon for a "missing" file that is "unavailable to the user, such as when a system that is hosting the file of interest is offline," is "presented as a ghosted icon that includes a small image of a red 'X' or other suitable visual indicia that the file is unavailable to the user." (*Id*., ¶79; Balakrishnan, ¶288.)

When obviously applied to Sigurdsson's system, the "ghosted" icon and red "X" disclose **availability of the version of the first file presented at the second client device based on the first metadata**, which would have been obvious to display on the "second client device" in Sigurdsson's system (e.g., Client C) to beneficially indicate to the user the availability of the file represented by the icon, before the device has downloaded the file itself, which could take a long time as discussed for [1d]. (Balakrishnan, ¶¶213-215.)

The same motivations to combine Sigurdsson with Shappell and Rao described above for [1d] apply fully here. (Balakrishnan, ¶212.) Sigurdsson contemplates standard "home" and "work" computers with Microsoft software but does not expressly describe GUI features. (Sigurdsson, ¶¶27, 42, 45, 58, 60;

Balakrishnan, ¶211.) Shappell picks up where Sigurdsson leaves off, motivating a POSA to consider GUI features to improve upon Sigurdsson with Shappell's teachings to graphically indicate to Sigurdsson system users the availability of modified files, with full expectation of predictable success. (Balakrishnan, ¶211.)

4. Claim 5.

(a) [5pre]-[5a]

Sigurdsson and Shappell disclose and render obvious [5pre]-[5a]. (Balakrishnan, ¶¶220-225.) Sigurdsson's clients are fungible—they generally "contain similar components and can accept and transmit content in a similar manner." (Sigurdsson, ¶¶50-51, 26-28 (user may work on a given file, switching between home and work computers); Balakrishnan, ¶221.) Clients A and C can both similarly post/push files to Server 106. (Sigurdsson ¶¶51, 32; Balakrishnan, ¶221.) Thus, as explained above for [1b] but in the **reverse direction**, Sigurdsson renders obvious [5pre]-[5a] regarding Client C ("**second client device**"). (Balakrishnan, ¶221.) As one exemplary "**second file**," Sigurdsson discloses an "office memo on the Client C." (Sigurdsson, ¶58; Balakrishnan, ¶221.)

(b) [5b]

Sigurdsson discloses: "***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." (Sigurdsson ¶61; Balakrishnan, ¶¶222.) This notification sent from client to server allows the server to "**determine that the server system is in**

communication with the first client device associated with the user." (Balakrishnan, ¶222.) For the reasons above regarding [1b] and [5a], Sigurdsson's clients all operate similarly, and it would have been obvious to determine that Client A ("**first client device**"), or any other client, is in communication with the server. (Balakrishnan, ¶¶222-223.)

(c) [5c]

As explained above for [5pre]-[5a], Sigurdsson's clients are fungible, and it would have been obvious to a POSA to also perform the functionality described above for claim 1 with respect to Clients A and C, further described in paragraphs 72-74 and 79-80, in the **reverse direction**. (Sigurdsson, ¶¶50-51, 72-74, 79-80; Balakrishnan, ¶224.) Client C would send the "**second file**" to the server, which would determine that it is in communication with Client A (after Client A has been offline, for example), send Client A the "list identifying the missing content," and, as send the missing content (i.e., the "**second file**") to Client A. (Sigurdsson, ¶¶73, 79-80; Balakrishnan, ¶224.)

Transferring the second file to Client A would occur **automatically**, with no user intervention. (Balakrishnan, ¶224.) In addition, the process would be "**responsive to (i) determining that the server system is in communication with the first client device and (ii) receiving the copy of the second file from the second client device**," because the server would need to check to make sure that

Client A is online and that it actually has a copy of the second file before attempting to send it. (Balakrishnan, ¶224.)

Moreover, given the fungibility of Sigurdsson's clients, it would have been obvious to receive a file from Client C at the server and then automatically transfer it to Client A in view of the embodiment described in connection with Figure 5, which describes a similar data flow between Clients A and B. (Sigurdsson, ¶¶52-55 ("In step 508, the server stores the content received from the originating client, and the content is propagated to the second client."); Balakrishnan, ¶224; *Boston Sci.*, 554 F.3d at 991 ("Combining two embodiments disclosed adjacent to each other in a prior art patent does not require a leap of inventiveness."); *Power Integrations*, 843 F.3d at 1337 (prior art "that sometimes, but not always, embodies a claimed method nonetheless teaches that aspect of the invention").)

For the same reasons explained for [1f], and obviously also applicable to the **reverse direction** (i.e., involving the second file modified at the second client device) as explained for [5pre]-[5a], a POSA also would have found it obvious for the server system to **automatically transfer the copy of the second file to replace an older version of the second file stored on the first client device** (Client A), both in view of Sigurdsson and a POSA's knowledge to avoid endlessly accumulating old outdated versions of files and as taught and motivated by Shappell's "automatic replication" feature. (Sigurdsson, ¶¶26-28, 84-85; Shappell,

¶¶62, 81, Figs. 19-20; Balakrishnan, ¶¶224-225.)

5. Claim 7.

Sigurdsson discloses and renders obvious claim 7's limitations. (Balakrishnan, ¶¶226-227.) Client A "posts" the copy of the file, also called a "***push***," responsive to the user modifying its content: "The Client A 102 ***posts*** Client A Content 108 to the Server 106 when content is changed on the Client A 102. . . . In this connection-oriented architecture . . . , each of the clients can ***push*** new data . . . ." (Sigurdsson, ¶¶84, 54 ("the HTTPS Client/server 326 of the Client A 102 can transmit an HTTP post command containing the Client X Content 410 to the Server 106."); Balakrishnan, ¶227.) A POSA would also understand that HTTP POST discloses a "push request." (Balakrishnan, ¶227.) It would have been obvious to incorporate this functionality associated with Figure 7 from Sigurdsson into other embodiments in Sigurdsson, such as the embodiment shown in Figure 6 and discussed above, given that they reference the same components (Server 106 and Clients A, B, and C) and the clients are fungible, as discussed above. (Sigurdsson, ¶¶50-51; Balakrishnan, ¶227.) *Boston Sci.*, 554 F.3d at 991 (obvious to combine adjacent embodiments).

6. Claim 11.

Claim 1 recites a system that executes the same method steps of claim 11, which is obvious for the same reasons. (Balakrishnan ¶401.)

7. Claim 12.

Sigurdsson with Shappell further renders obvious claim 12. (Balakrishnan ¶¶402-408.) Sigurdsson discloses a "sequence diagram showing an illustrative method for sharing and synchronizing content" using the "server architecture of FIG. 4 ***when a client that is offline comes online***." (Sigurdsson ¶61; Balakrishnan ¶405.) "In this case, ***the system 400 automatically detects that a particular client is offline***. 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***." (Sigurdsson ¶61; Balakrishnan ¶405.) For example, Sigurdsson discloses determining whether "Client C" is offline (meaning that Client C would not be in communication with the server). (Sigurdsson ¶66; Balakrishnan ¶406.) As explained above, Sigurdsson's various "clients" are fungible, so the same functionality would be applied with respect to Clients A, B, or C in Sigurdsson. (Sigurdsson ¶51 ("The Server 106 can still serve as a drop-box for temporarily (or permanently) storing Client X Content 410 originating from ***any the*** [sic] ***Clients 102, 104 or 406***. ***The drop-box, or storage, capabilities of the Server 106 can be used, for example, if a particular client is offline when the first attempt***

is made to deliver new content to it. Although FIG. 4 does not show explicit arrows between the Client C 406 and the Server 106, the Client C 406 can have communicate [sic] with the Server 106 in a manner similar to the method used by the Clients A or B."); Balakrishnan ¶406.) Thus, Sigurdsson discloses and renders obvious **initially determining, by the server system, that the server system is not in communication with the second client device associated with the user**, followed by **resuming communication with the second client device** and **determining that the server system is in communication with the second client device**. (Balakrishnan ¶¶403-406.)

Sigurdsson further describes 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 ¶436.) The transfer of "Client X Content 410" can be sent automatically to a client that was temporarily offline when it comes back online. (Sigurdsson, ¶69 ("Step 612 can occur ***after the third client comes online, having been offline during the time that the content originated on the first client and was propagated to the second client and to the server***. In step 612, the third client requests a list of missing content from the server. For example, ***the Client C 406 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."); Balakrishnan ¶436.) "In step 614, the server determines the content missing from the third client." (*Id.* ¶70.) "In step 616, ***the server passes the list of missing content to the third client*** . . . , which had been offline." (*Id.* ¶73.)

A POSA would have found this claim element to be obvious with respect to Clients A and C as described in Sigurdsson Figure 6 and paragraphs 72-74 and 79-80. (Sigurdsson, ¶¶72-74, 79-80, Fig. 6; Balakrishnan, ¶366.) Namely, Client A would send the "**first file**" to the server, which, in turn, would determine that it is in communication with Client C (after Client C has been offline and resumed communications, for example). (Balakrishnan, ¶366.) Thereafter, the server would send Client C the "list identifying the missing content" referenced in paragraph 73 and, as described in paragraph 79-80, send missing content (i.e., the "**first file**") to Client C. (Sigurdsson, ¶¶73, 79-80; Balakrishnan, ¶366.)

Transferring the first file to Client C would occur **automatically**, with no user intervention. (Balakrishnan, ¶367.) Moreover, given the fungibility of Sigurdsson's clients, it would have been obvious to receive a file from Client A at the Server and then automatically transfer the file to Client C in view of the embodiment described in connection with Figure 5, which describes a similar data flow between Clients A and B. (Sigurdsson ¶¶52-55 ("In step 508, the server stores the content received from the originating client, and the content is propagated to the second client."); Balakrishnan, ¶367.)

As discussed for [1f], Sigurdsson's discussion of synchronizing versions of a file between a user's multiple computers would have motivated a POSA to consider the obvious solution to **replace** an older version of the modified file at the receiving computer, which Shappell further renders obvious as discussed for [1f]. (Balakrishnan, ¶¶224-225, 348.) Shappell's "automatic replication" option, discussed for [1f], further renders obvious implementing Sigurdsson's Server 106 to **automatically transfer the copy of the first file to the second client device to replace the older version of the first file stored on the second client device with the copy of the first file, responsive to (i) resuming communication with the second client device, (ii) determining that the server is in communication with the second client device, and (iii) receiving the copy of the first file from the first client device**. (Balakrishnan, ¶¶224-225, 348.) For the same reasons as explained for [1f], it would have been obvious to implement Shappell's "automatic replication" feature as an option for each Sigurdsson client device, including Client C ("**second client device**"), so that Server 106 would **automatically** transfer to Client C the copy of a file modified at Client A (the "**first client device**"), responsive to Server 106 receiving the copy of the modified file from Client A and Client C coming back online and resuming communication with a request for the list of missing content (which would include the modified file from Client A), and replacing the older

version of the file at Client C. (Sigurdsson, ¶¶69-74, 79-80; Balakrishnan, ¶¶224-225, 348.)

8. Claim 14.

Claim 14 is substantively identical to claim 3, and obvious for the same reasons. (Balakrishnan ¶409.)

9. Claim 16.

Claim 16 is substantively identical to claim 1, except it adds **"[o]ne or more non-transitory machine-readable media storing instructions that, when executed by one or more processors of a server system, cause operations**." Sigurdsson discloses that Server 106 can include "System 900" in Figure 9. (Sigurdsson, ¶104; Balakrishnan ¶236-239.) System 900 includes "a ***Memory 920***" and a "***Storage Device 930***." (Sigurdsson, ¶105; Balakrishnan, ¶237.) Memory **920** may be a "non-volatile memory unit" and "***[s]torage devices suitable for tangibly embodying computer program instructions and data include all forms of non-volatile memory . . .*** ." (Sigurdsson, ¶¶105-106, 110; Balakrishnan, ¶237.) Thus, this claim would have been obvious over Sigurdsson, Shappell, and Rao for these reasons and those above for claim 1. (Balakrishnan, ¶237.)

10. Claim 17.

Claim 17 is substantively identical to claim 3, and obvious for the same reasons. (Balakrishnan ¶411.)

C. Ground 2: Obviousness over Brown with Hesselink, Shappell, and Rao.

1. Claim 1.

(a) [1pre]

Brown discloses a client/server **system** shown in Figure 1, including client devices (such as desktop **130** and portable **135**) and server **105**. (Brown, Fig. 1, 2:57-3:16, 3:54-4:28; Balakrishnan, ¶¶577-580.)

(b) [1a]

Brown discloses and renders obvious [1a]. (Balakrishnan, ¶¶581-583.) The **"server system"** is server **105**, including Synchronization File System (SFS) **110**. (Brown, 3:56-60, 2:59-62, Fig. 1; Balakrishnan, ¶582.) A POSA would have understood Brown's server includes and obviously would include **one or more processors programmed with computer program instructions that are executed to cause** the operations performed by the server, as it "includes all the typical elements of a server, such as a central ***processor***, ***memory***, bus, disk space, etc." (Brown, 3:57-59; claims 38-39; Balakrishnan, ¶583.)

(c) [1b]

Brown with Hesselink discloses and renders obvious [1b]. (Balakrishnan, ¶¶584-611.) The "**first client device associated with a user**" is a Brown client device, such as desktop computer **130**. (*Id.*, ¶585.) Brown discloses and renders obvious multiple client devices, each associated with the same user, addressing a

perceived need in the prior art to synchronize files between a user's multiple computers (e.g., home and work). (Brown, 1:24-50, 4:20-22, 4:62-64 ("multiple clients of a single user account"), 11:55-56 ("each user can have several clients accessing a single folder"); Balakrishnan, ¶585.) Brown describes synchronization with reference to client **130** while describing that "***[c]lient machines synchronize with each other by synchronizing to a common server account***," teaching that the client devices associated with a user (such as devices **130** and **135**) are fungible insofar as files are synchronized bi-directionally between client devices **130** and **135** as files are modified on either device. (Brown, 2:65-67, Figs. 4A-4C, 10:7-9, 10:7-11:52, Figs. 8A-11F, 13:57-16:39 ("synchronizing the clients and server of FIG. 1"), 14:7-9, 14:60-62, 15:30-32; Balakrishnan, ¶586.)

Brown in view of Hesselink discloses and renders obvious that server **105** is configured to **receive, over a network, a copy of a first file from a first client device associated with a user, wherein the copy of the first file is automatically received from the first client device responsive to the user modifying a content of the first file stored on the first client device, the copy of the first file being a version of the first file that is generated from the user modifying the content of the first file**. (Balakrishnan, ¶588.)

Brown describes the **user modifying a content of a first file stored on the first client device** (e.g., computer **130**), generating an "updated" version of the file.

(Brown, 11:64-66; Balakrishnan, ¶589.) As a result of the modification, server **105 receives, over the network, a copy of the file** when more than a threshold portion of the file has changed, based on an algorithm. (Brown, 11:22-30, 12:15-16, 12:41-51 (if less than a threshold portion of the modified file matches the original file, "***the entire file is transmitted***"), Fig. 11B (steps 1117, 1122, 1120) 15:41-54 (at step **1120** the client "uploads the entire file"); Balakrishnan, ¶589.) *Power Integrations*, 843 F.3d at 1337 (prior art teaches limitation where it "sometimes, but not always" implements it).

Brown discloses "automatic" synchronization of modified files: "From the user's perspective, ***the synchronization process is automatic***, runs in the background, and requires minimal monitoring or invention." (Brown, 3:40-42; Balakrishnan, ¶592.) File system activity on the client is monitored "to determine which files on the client have been changed" and "locate changes to send to the server." (Brown, 3:47-50, 8:13-17; Balakrishnan, ¶595.) To synchronize those monitored modifications of files, one "embodiment" "initiates synchronization on a fixed time interval," but alternatively the system "can use any scheduling algorithm to initiate synchronization," and the user may manually initiate synchronization "at any time." (Brown, 3:42-47; Balakrishnan, ¶596.)

Hesselink further discloses and renders obvious the first client transferring a copy of a file specifically "**responsive to the user modifying a content**" of the file.

(Balakrishnan, ¶¶597-600.) Like Brown, Hesselink teaches "synchronization of files among a user's multiple computers," such as "home" and "work" computers. (Hesselink, ¶¶210, 192; Balakrishnan, ¶597.) When the user modifies/edits file content, the system automatically transmits the changes to the file over the network. "***As the user works on the file***, such as by ***adding to the file***, ***modifying or editing***, each time that the user saves his/her work, the system compares the differences between the saved version of the file and the previous version of the file, and, as long as the user module remains connected to the network, the changes/differences are ***transmitted over the network to the remote device module*** . . . ." (Hesselink, ¶189; Balakrishnan, ¶598.) The synchronization occurs "***as soon as the update has been performed*** at one location," so that "all computers" in the network "may be automatically updated concurrently with the updates . . . each time an update is performed." (Hesselink, ¶¶183, 189; Balakrishnan, ¶¶598-599.) This automatic-update function optionally "may be set to apply to only one file or any number of files." (Hesselink, ¶210; Balakrishnan, ¶600.)

Motivation to Combine Brown with Hesselink

It would have been obvious to combine the teachings of Brown and Hesselink, predictably resulting in Brown's synchronization system in which the scheduling

algorithm causes the upload of a modified file from a user device responsive to the user modifying the file as taught by Hesselink. (Balakrishnan, ¶¶601-611.)

Brown and Hesselink are analogous references in the same field as the '622, directed to synchronizing files across a user's devices. (Balakrishnan, ¶602.) The combination would have straightforwardly combined prior art elements according to known techniques to yield a predictable improvement to Brown's system, by synchronizing files from a client device to Brown's server responsive to the user modifying the file. (Balakrishnan, ¶603.)

Hesselink and Brown both motivate the combination. (Balakrishnan, ¶¶604-611.) A POSA would have been motivated to consider Hesselink's teachings to serve Brown's stated goal to provide "a way to maintain distributed files across multiple clients, ***maintaining currency at each client as changes are made***," and Brown expressly motivates consideration of "any scheduling algorithm to initiate synchronization." (Brown, 1:47-50, 3:44-46; Balakrishnan, ¶604.) Hesselink provides complementary express motivations for a POSA to integrate Hesselink's teachings into Brown's systems, including (1) saving time for the user, and (2) avoiding the problem of having two different versions, or an outdated version, of a file being edited by a user: "***This functionality saves the user a great deal of time***, since there is no need to synchronize files when the user returns to the site of the device module. ***Further, there is no concern over having two different versions of***

a file, or concern that a user is not using the latest, most up to date version of a file." (Hesselink, ¶189; Balakrishnan, ¶605.)

A POSA would have reasonably expected success in implementing this modified Brown system, with implementation well within the grasp of a POSA. (Balakrishnan, ¶607.) A POSA would not have been dissuaded by differences between Brown and Hesselink, given the compelling express motivations. (Balakrishnan, ¶608.) For example, a POSA would not have been dissuaded by Hesselink's disclosure of transmitting only the changes to a file to conserve bandwidth, as Brown already teaches that either only the changes, or the entire file, can be transferred, providing a more nuanced approach that balances tradeoffs. (Balakrishnan, ¶608.) Hesselink also motivates that its automatic-change-synchronization "may be set to apply to only one file or any number of files" at the user's option (Hesselink, ¶210), further mitigating any concern about overall bandwidth usage, complementing Brown's disclosure to synchronize only files within a certain designated directory/folder on each client. (Brown, 6:61-7:20; Balakrishnan, ¶610.) Nor would a POSA have been dissuaded by Hesselink using peer-to-peer rather than client-server communication, as Hesselink does not categorically disparage client/server models and contemplates that each synchronized computer can be a "file server." (Hesselink, ¶¶23-34, 122; Balakrishnan, ¶611.)

(d) [1c]

Brown discloses and renders obvious [1c]. Brown's server stores a copy of the uploaded modified file and maintains files in storage to be downloaded by other clients. (Brown, 12:53-13:3 (once modified file is uploaded, "the server then stores the file"), Abstract, 3:54-4:3; Balakrishnan, ¶751.)

(e) [1d]

Brown with **Shappell** and **Rao** disclose and render obvious [1d] and the priority-related limitations of claims [1e] and 3, addressed together as in Ground 1. (Balakrishnan, ¶¶611-645.)

Brown discloses and renders obvious (and renders obvious with Shappell and Rao) the priority-related limitations of claims [1d], [1e], and 3 at least to the same extent the '622 specification discloses them. (*Id.*) Brown's server **receives, from the first client device** (e.g., computer **130**), **metadata associated with the modified file that is generated from the user modifying the content of the file**, and Brown discloses and renders obvious that this metadata associated with a modified file is **automatically transferred with greater priority than the file itself** between the client and server. (Balakrishnan, ¶613.)

Brown describes various **metadata**, including update time, file length/size, IDs, and other data generated from "file system activity," which includes the user modifying file content. (Brown, 5:25-6:10, 6:61-7:5, 7:21-43; Balakrishnan, ¶¶613-

616.) During file synchronization, the client transfers updated metadata to the server. (Balakrishnan, ¶614.) Brown specifically describes that **metadata** for a modified file is uploaded from the originating "**first client device**" to the server (e.g., "CSI," *see* Brown, 10:9-10, "SID" ("Server ID"), "parent directory SID," and "file name," *id.*, 15:38-40, 5:50-52) prior to the modified file being uploaded. (Brown, 12:53-13:3 (upload SID prior to modified file), 13:57-62 and Fig. 8A (step **805**, upload CSI prior to modified file), 14:5-6 and Fig. 8B (step **850**, upload changes, detailed at Figs. 11A-C), Figs. 11A-C and 15:30-54 (step **1115** upload SID, parent directory SID, file name; step **1120** upload modified file); Balakrishnan, ¶¶614-619.) Accordingly, the system is configured so that the server **receives, from the first client device, first metadata associated with the modified version of the file, before receiving the copy of the file itself**. (*Id.*)

Then, **before** the server transfers a copy of the modified file to a "**second client device**" (another synchronizing client, e.g., computer **135**), the server first transfers to that receiving client the Server Synchronization Data ("SSD") that contains the **metadata** of all files "***that have been changed***" since the last synch by the client. (Brown, 5:25-49, 10:19-27, 8:48-61 (describing SSD contents including SID), 2:65-67 ("Client machines synchronize with each other by synchronizing to a common server account."); Balakrishnan, ¶¶616, 681.) Figures 8B and 9A show the server-to-client synchronization process including (1) step **825**, client receives SSD

(including the file metadata) from the server, followed by (2) step **915**, client downloads the file from the server, which is further described in Figures 10A-10B for a modified file, including step **1050** where an entire modified file is downloaded from the server. (Brown, Figs. 8B (step **825**), 9A (step **915**), 10A-10B (step **1050**), 13:65-14:3, 14:7-19, 14:60-15:16; Balakrishnan, ¶¶616-619.)

To the extent the "priority"-related limitations of claims [1d], [1e], and 3 are interpreted to require further categorical assignment of priorities to metadata and files, Brown in view of **Shappell** and **Rao** disclose and render them obvious for similar reasons as Ground 1 above. (Balakrishnan, ¶622.)

Brown teaches that it is "***desirable to minimize the amount of data a server sends to or receives from a client, so that other clients' requests can be handled in a timely manner***." (Brown, 11:60-63.) This teaching would have motivated a POSA to consider other references focused on sending metadata prior to file data to further improve upon Brown's approach for synchronizing files using available network resources efficiently. (Balakrishnan, ¶¶623-645.)

Shappell and **Rao** render obvious a predictable variation where Brown's clients and/or server categorically assign greater priority to metadata and lower priority to files (**priority assignment configuration**) and transmit metadata prior to associated files based on such prioritization, to advantageously inform clients about updated files. (Balakrishnan, ¶623.)

For substantially the same reasons above for Ground 1, **Shappell** and **Rao** disclose and motivate this obvious modification to Brown. Like Brown, **Shappell** teaches synchronizing files across multiple computers using metadata. (Shappell, ¶9; Balakrishnan, ¶627.) As discussed above, Shappell teaches that a file's metadata is transferred first (with a higher priority) to all connected group computers before sending a copy of the file itself, to facilitate user-interface display of the files' availability. (Balakrishnan, ¶¶628-630.)

For the reasons provided above for Ground 1, **Rao** provides naturally complementary teachings to categorically prioritize synchronizing metadata before files between server and client, to advantageously enable functions at the client. (Balakrishnan, ¶631.) Like Brown, Rao describes file synchronization between a client and a server over a network. (Rao, ¶¶26, 37, 43, 63; Balakrishnan, ¶631.)

Motivation to Combine Brown, Hesselink, Shappell, and Rao

Brown, Hesselink, Shappell, and Rao are analogous references in the same field as the '622, directed to synchronizing files and associated metadata. (Balakrishnan, ¶¶632-645.) The combination would have straightforwardly combined prior art elements according to known techniques to predictably improve the Brown/Hesselink system in the same way the relevant features enhance Shappell's and Rao's disclosed systems, by automatically transmitting metadata as a priority based on file changes to timely notify users of an updated version of a file

and enable functions (e.g., displaying information about a modified file) at the receiving client. (Balakrishnan, ¶634.)

A POSA would have been motivated to look to Shappell and Rao to predictably improve Brown's synchronization system with Shappell and Rao's metadata-priority features. (Balakrishnan, ¶624.) A POSA would have recognized that Brown's disclosed synchronization process leaves open the possibility that the user of one client device (for example, computer **130**) may have modified a certain file, but a second client device connected to the network that has an earlier version of the same file (such as client device **135**) will not immediately receive the modified file. (Balakrishnan, ¶624.) That raises the problem that the user at the second client device might want to work with that file but would not know whether the local copy of the file was up-to-date. (Balakrishnan, ¶624.)

Significantly, even where Brown's system is modified in view of Hesselink so that each client transfers a copy of a modified file to the server as the user modifies the file (where Brown's file-changes-threshold is exceeded), another client device might not receive a copy of the modified file until after a substantial amount of time has passed, leaving the user uninformed about the modification, such as where a relatively large file is being transferred over a slow connection. Brown explains: "***if a user has a slow network connection . . . it can be time-consuming to have to wait for the entire document to upload or download***." (Brown, 12:8-12; Balakrishnan,

¶624.) Rao confirms that large file transfers could take "many minutes or even several hours." (Rao, ¶48; Balakrishnan, ¶636.) A POSA would have recognized that Brown's system thus gives rise to a potential gap in synchronicity between versions of a synchronizing document. (Balakrishnan, ¶636.) For example, the user of a synchronizing client device (e.g., work computer) in Brown may work on modifying the large file. (*Id.*) When the user subsequently accesses another synchronizing client device (for example, home computer) to continue work on that file, the user would benefit from being able to quickly and visually determine whether or not the version of the file on the home computer is current, before opening the file to work on it. (*Id.*)

A POSA would have looked to references like Shappell and Rao to address this issue, predictably resulting in the Brown/Hesselink system with a priority assignment configuration where the metadata associated with a file modified at the originating first client device (for example, device **130**) is transmitted automatically by that device to the server, and then by the server **to a second client device** (for example, device **135**), **with a first priority that is higher than a second priority that is assigned to files**, thereby enabling a graphical availability indication at the second client device, advantageously promptly notifying the user at that device of the updated (but still not yet downloaded) file across the network. (Balakrishnan, ¶636.)

A POSA also would have been motivated to consider both Shappell and Rao given their originations with file-management leader Microsoft, as Brown likewise contemplates standard "home" and "work" computers with Microsoft software. (Brown, 1:25-29, 7:6-11; Balakrishnan, ¶641.) A POSA also would have appreciated the benefits of prioritizing metadata as taught by Boyce, as explained for Ground 1.

A POSA would have reasonably expected success in implementing this modified Brown system, as Brown already discloses metadata and file transfer using standard, familiar computer, OS, and network components. (Balakrishnan, ¶643.) For similar reasons as Ground 1, a POSA would not have been dissuaded by Shappell's primary embodiment being peer-to-peer rather than client-server, nor by Shappell's example embodiment "typically" having multiple users of devices (Shappell, ¶40), given Shappell's compelling teachings and motivations to improve Brown's system. (Balakrishnan, ¶189.)

Hesselink also would not dissuade a POSA from combining the above-discussed features. (Balakrishnan, ¶¶644-645.) Hesselink discusses computers running Microsoft Windows (Hesselink ¶¶78, 187, 203), a connection server **14** running Windows NT (*id.*, ¶80), and Windows Explorer "to access remote storage" (*id.*, ¶¶127-128, 133-134). (*Id.*, ¶644.) A POSA would have viewed Microsoft references like Shappell and Rao, which discuss Microsoft GUIs, as complementary,

particularly as Hesselink discusses Windows Explorer's use of "file overhead information" (i.e., metadata) "***to list or display directory data structures for user navigation purposes***." (Hesselink, ¶134; Balakrishnan, ¶644.) Hesselink also discusses optional "subscriptions" for metadata ("file overhead" data) and file data, further supporting a POSA's consideration of references like Shappell and Rao for the benefits of prioritizing transfer of metadata over file data to facilitate graphical indications of file availability. (Hesselink, ¶136; Balakrishnan, ¶644.)

(f) [1e]

Brown with Shappell and Rao discloses and renders obvious [1e] for the reasons discussed for [1d] above. (Balakrishnan, ¶585.)

(g) [1f]

Brown and Hesselink render obvious [1f]. Brown's client devices synchronize bidirectionally "with each other by synchronizing to a common server account," with "**automatic**" synchronization. (Brown, 2:65-67, 3:40-42; Balakrishnan, ¶561.) Responsive to receiving a modified file from one client device (e.g., computer **130**), the server automatically (without user intervention) transfers that file over the network to another synchronizing client device (e.g., computer **135**) associated with the same account. (Brown, 10:18-27, 10:53-55 (describing that when the server has changes the client lacks, then the client will download all the pertinent changes from the server); Balakrishnan, ¶755.) The downloaded modified

file "**replac[es] an older version of the file** . . . ." (Brown, 13:18-23; Balakrishnan, ¶756.)

Hesselink further renders obvious implementing Brown's server to **automatically** transfer copies of modified files to connected clients responsive to receiving them. (Balakrishnan, ¶¶756-761.) As discussed for [1b], a POSA would have been motivated to consider Hesselink's automatic-update teachings to serve Brown's goal of "***maintaining currency at each client as changes are made***" using "any scheduling algorithm to initiate synchronization." (Brown, 1:46-50, 3:44-46; Balakrishnan, ¶758.) As discussed for [1b], Hesselink teaches to "***automatically update***" connected computers with modifications to files, including currently connected computers and computers that "go online again" after being offline. (Hesselink, ¶¶183, 190, 197; Balakrishnan, ¶760-761.) The same motivations to combine Brown and Hesselink discussed for [1b] apply. (Balakrishnan, ¶¶601-611.)

2. Claim 3.

As explained for [1d], Brown with Shappell and Rao discloses and renders obvious claim 3.

3. Claim 4.

Shappell renders obvious claim 4's limitations. (Balakrishnan, ¶764.) As discussed for [1d] and for claim 4 in Ground 1, Shappell teaches that when a file is modified at one computer, metadata is transmitted and received by other computers,

causing them to change the icon for that file to a "ghosted" or semi-transparent icon indicating an "available" file, which may add a red "X" when that file is currently "unavailable to the user." (Shappell, ¶¶79, 81, Figs. 19-20 (steps 1902, 1904, 1906); Balakrishnan, ¶¶612-647, 664-667.) When obviously applied to the Brown/Hesselink system, the "ghosted" icon and red "X" disclose **availability of the version of the first file presented at the second client device based on the first metadata**, which would have been obvious to display on the "second client device" (e.g., computer **135**) to beneficially indicate to the user the availability of the file represented by the icon, before the device has downloaded the file itself, which could take a long time as discussed for [1d]. (Balakrishnan, ¶¶612-647, 664-667.)

The same motivations to combine Brown and Hesselink with Shappell and Rao described above for [1d] apply fully here. (Balakrishnan, ¶¶632-645.) Brown and Hesselink disclose computers running OSs such as Windows with GUIs, but do not detail the GUIs. (*E.g.*, Brown, 7:6-11; Hesselink, ¶187; Balakrishnan, ¶663.) Shappell picks up where they leave off, motivating a POSA to consider GUI features to improve upon the Brown/Hesselink synchronization system with GUI features to graphically indicate to Sigurdsson system users the availability of modified files, with full expectation of predictable success. (Balakrishnan, ¶663.)

4. Claim 5.

(a) [5pre]-[5a]

As discussed above for [1b], Brown's client devices (e.g., clients **130** and **135** in Figure 1), obviously associated with the same user, are fungible insofar as they communicate with the server in the same way, synchronizing modifications to multiple files on each client device bi-directionally through the server. (Brown, 2:65-67; Balakrishnan, ¶¶674-686.) Brown describes a given client both uploading a modified file and downloading a file modified by another ("second") client (*e.g.*, *id.*, 12:52-13:23). (Balakrishnan, ¶¶584-645.) Thus, for the same reasons as [1b], this limitation would have been obvious to apply to a "second" file modified at the "second" client. (Balakrishnan, ¶¶584-645.)

(b) [5b]

Brown with Hesselink discloses and renders obvious [5b]. (Balakrishnan, ¶¶676-678.) For example, it would have been obvious that when synchronizing the server with each Brown client device, the client would initiate communication with the server, and the server would respond based on a determination that client and server are in communication, and obviously using well-known handshake type functions in TCP/IP or HTTP or other protocols, as Brown teaches using TCP/IP and HTTP (or "other protocols"): "The client SA communicates with the server SFS via TCP/IP. 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. ***Communication between the client SA and the server SFS is initiated by the client SA and responded to by the server SFS***." (Brown, 3:1-8; Balakrishnan, ¶677.)

Hesselink further renders obvious this element, disclosing determining whether computers are online/offline and, if offline, storing modifications for later transmission, with offline computers "automatically updated as soon as those computers go online again." (Hesselink, ¶¶183, 197 (after "re-establish a connection," "updates (including any updates to file data . . .) may be automatically transmitted" by a user module); Balakrishnan, ¶678.) A POSA would have been motivated to consider Hesselink's teachings to serve Brown's goal of "***maintaining currency at each client as changes are made***," with Brown encouraging "any scheduling algorithm to initiate synchronization." (Brown, 1:47-50, 3:44-46; Balakrishnan, ¶686.) The same motivations to combine Brown and Hesselink discussed above for [1b] apply equally here. (Balakrishnan, ¶¶601-611.)

(c) [5c]

Brown with Hesselink discloses and renders obvious [5c]. As discussed for [1b], [1f], and [5pre]-[5a], Brown's clients are functionally interchangeable, bi-directionally "synchronizing to a common server account" with "**automatic**" synchronization. (Brown, 2:65-67, 3:40-42; Balakrishnan, ¶¶679-686.) Responsive to Brown's server determining that it is in communication with the first client device

(e.g., computer 130) per [5b], the server automatically (without user intervention) transmits to that first client device the second file received from the second client (e.g., computer 135) per [5a], which "**replace[s] an older version of the file**" (*Id.*, 13:18-23), using Brown's automatic synchronization process discussed for [1b] and [1f]. (Balakrishnan, ¶¶679-686.) As further discussed for [1b] and [1f], Hesselink further renders obvious implementing Brown's server to **automatically** transmit modified files to connected clients responsive to (i) determining that it is in communication with clients, and (ii) receiving the modified files. The same disclosures and obviousness rationale for [1b] and [1f] apply equally here for synchronization of a modified file in the **reverse direction** from the "second" client (e.g., computer 135) through the server to the "first" client (e.g., computer 130). (Balakrishnan, ¶¶601-611.)

5. Claim 7.

Brown and Hesselink disclose and render obvious claim 7's limitations. (Balakrishnan, ¶¶687-692.) Brown's client Synchronization Applications (SAs) "initiate" communication that the server SFS "respond[s] to." (Brown, 2:59-3:8; Balakrishnan, ¶690.) Brown's clients submit "***push***" requests to the server. (Brown, 4:62-64 (clients "push changes to the server"), 15:30-35, Abstract (clients "***push*** local changes of the files onto the server"); Balakrishnan, ¶691-692.) As explained

for [1b], Hesselink renders obvious Brown's clients automatically pushing changed files responsive to user file modifications. (Balakrishnan, ¶¶597-600.)

6. Claim 11.

Claim 1 recites a system that executes the same method steps of claim 11, which is obvious for the same reasons. (Balakrishnan ¶790.)

7. Claim 12.

Brown and Hesselink further render obvious claim 12. As discussed for [5b], Brown discloses and renders obvious determining whether client and server are in communication, such as using protocol handshakes. (Brown, 3:1-8; Balakrishnan, ¶¶791-798.) Brown discloses that in its "automatic" synchronization process, including as obviously modified in view of Hesselink, the downloaded modified file **replaces an older version of the file**, as discussed for [1f]. (*Id.*, 13:18-20, 3:40-42; Balakrishnan, ¶¶679-686.) As also discussed for [5b], Hesselink further renders obvious determining whether computers are online/offline and, if offline, storing modifications for later transmission, with offline computers "automatically updated as soon as those computers go online again," with the system "automatically and repeatedly" attempting to reconnect until it can be "automatically updated," which is obviously "useful" for broken or suspended network connections. (Hesselink, ¶¶183, 190, 197 (after "re-establish a connection," "updates (including any updates to file data . . .) may be automatically transmitted" by a user module); Balakrishnan,

¶760-761.) As discussed previously, POSA would have been motivated to consider Hesselink's teachings to serve Brown's goal of "***maintaining currency at each client as changes are made***," with Brown encouraging "any scheduling algorithm to initiate synchronization." (Brown, 1:46-50, 3:44-46; Balakrishnan, ¶686.) The same motivations to combine Brown and Hesselink discussed above for [1b], [1f], and [5b] apply equally here. (Balakrishnan, ¶¶601-611.)

Thus, Brown with Hesselink discloses and renders obvious [12a] **initially determining, by the server system, that the server system is not in communication with the second client device associated with the user** (applying Hesselink's teaching of "repeatedly" checking if connected computers are offline or online), and [12b] **automatically transferring, by the server system** (as disclosed by Brown's "automatic" synchronization and as modified in view of Hesselink), **the copy of the first file to the second device to replace** (as taught by Brown) **the older version of the first file stored on the second client device with the copy of the first file, responsive to (i) resuming communication with the second client device and (ii) determining that the server system is in communication with the second client device** (applying Hesselink's teachings for resuming communication with a previously-offline device with Brown's "automatic" synchronization process whereby server and clients communicate), **and (iii) receiving the copy of the first file from the first client device** (the modified file uploaded to Brown's server from

Brown's first client device, e.g., computer 130, implementing Brown's "automatic" synchronization including as obviously modified via Hesselink's automatic-update teachings per [1b]). (Balakrishnan, ¶¶791-798.)

8. Claim 14.

Claim 14 is substantively identical to claim 3, and obvious for the same reasons. (Balakrishnan, ¶799.)

9. Claim 16.

Claim 16 is substantively identical to claim 1, except it adds **"[o]ne or more non-transitory machine-readable media storing instructions that, when executed by one or more processors of a server system, cause operations**." Brown's SFS application is "[t]ypically" "installed on a hard disk drive within server 105" or "other forms of media or optical media." (Brown, 3:59-62; Balakrishnan, ¶¶701-703.) A POSA would have understood that a hard disk (or other media) obviously includes "[o]ne or more non-transitory machine-readable media storing instructions that, when executed by one or more processors of a server system, cause operations" on Brown's server. (Balakrishnan, ¶703.)

10. Claim 17.

Claim 17 is substantively identical to claim 3, and obvious for the same reasons. (Balakrishnan, ¶801.)

IX. CONCLUSION

Petitioners respectfully request IPR institution.

Dated: January 5, 2023

Respectfully submitted,

COOLEY LLP
ATTN: Patent Group
1299 Pennsylvania Avenue NW
Suite 700
Washington, DC 20004
Tel: (202) 842-7889
Fax: (202) 842-7899

By: /Heidi L. Keefe/
Heidi Keefe
(Reg. No. 40,673)
COOLEY LLP
ATTN: Patent Group
1299 Pennsylvania Ave., NW
Suite 700
Washington, D.C. 20004
Tel: (650) 843-5001
Fax: (650) 849-7400
hkeefe@cooley.com

Counsel for Petitioners

APPENDIX A

Claims Listing
Claims 1, 3-5, 7, 11, 12, 14, 16, and 17
U.S. Pat. No. 11,003,622

Claim 1:

[1pre] A system comprising:

[1a] a server system comprising one or more processors programmed with computer program instructions that, when executed, cause the server system to:

[1b] receive, over a network, a copy of a first file from a first client device associated with a user, wherein the copy of the first file is automatically received from the first client device responsive to the user modifying a content of the first file stored on the first client device, the copy of the first file being a version of the first file that is generated from the user modifying the content of the first file;

[1c] store the copy of the first file on the server system;

[1d] receive, from the first client device, first metadata associated with the version of the first file that is generated from the user modifying the content of the first file, the first metadata being assigned a first priority greater than a second priority assigned to the copy of the first file;

[1e] automatically transfer, based on the first priority being greater than the second priority, the first metadata to the second client device such that the first metadata is transferred to the second client device prior to the copy of the first file being transferred to the second client device; and

[1f] automatically transfer, over a network, the copy of the first file to the second client device associated with the user to replace an older version of the first file stored on the second client device, responsive to receiving the copy of the first file from the first client device.

Claim 3: The system of claim 1, wherein at least one of the server system or the first client device comprises a priority assignment configuration to assign greater priority to metadata associated with files than priority assigned to the files such that at least one of the server system or the first client device assigns the first priority to the first metadata and the second priority to the copy of the first file based on the priority assignment configuration.

Claim 4: The system of claim 1, wherein availability of the version of the first file is presented at the second client device based on the first metadata.

Claim 5:

[5pre] The system of claim 1, wherein the computer program instructions, when executed, cause the server system to:

[5a] receive a copy of a second file from the second client device associated with the user, wherein the copy of the second file is automatically received from the second client device responsive to the user modifying a content of the second file stored on the second client device, the copy of the second file being a version of the second file that is generated from the user modifying the content of the second file;

[5b] determine that the server system is in communication with the first client device associated with the user; and

[5c] automatically transfer the copy of the second file to the first client device associated with the user to replace an older version of the second file stored on the first client device, responsive to (i) determining that the server system is in communication with the first client device and (ii) receiving the copy of the first file from the second client device.

Claim 7: The system of claim 1, wherein the copy of the first file is automatically received from the first client device responsive to (i) a push request of the first client device and (ii) the user modifying the content of the first file stored on the first client device.

Claim 11:

[11pre] A method being implemented by a server system comprising one or more processors executing computer program instructions that, when executed, perform the method, the method comprising:

[11a] receiving, by the server system, over a network, a copy of a first file from a first client device associated with a user, wherein the copy of the first file is automatically received from the first client device responsive to the user modifying a content of the first file stored on the first client device, the copy of the first file being a version of the first file that is generated from the user modifying the content of the first file;

[11b] receiving, by the server system, from the first client device, first metadata associated with the version of the first file that is generated from the user modifying the content of the first file, the first metadata being assigned a first priority greater than a second priority assigned to the copy of the first file;

[11c] automatically transferring, by the server system, based on the first priority being greater than the second priority, the first metadata to the second client device such that the first metadata is transferred to the second client device prior to the copy of the first file being transferred to the second client device; and

[11d] automatically transferring, by the server system, over a network, the copy of the first file to the second client device associated with the user to replace an older version of the first file stored on the second client device, responsive to receiving the copy of the first file from the first client device.

Claim 12:

[12pre] The method of claim 11, further comprising:

[12a] initially determining, by the server system, that the server system is not in communication with the second client device associated with the user; and

[12b] automatically transferring, by the server system, the copy of the first file to the second client device to replace the older version of the first file stored on the second client device with the copy of the first file, responsive to (i) resuming communication with the second client device, (ii) determining that the server system is in communication with the second client device, (iii) and receiving the copy of the first file from the first client device.

Claim 14: The method of claim 11, wherein at least one of the server system or the first client device comprises a priority assignment configuration to assign greater priority to metadata associated with files than priority assigned to the files such that at least one of the server system or the first client device assigns the first priority to the first metadata and the second priority to the copy of the first file based on the priority assignment configuration.

Claim 16:

[16pre] One or more non-transitory machine-readable media storing instructions that, when executed by one or more processors of a first computer system, cause operations comprising:

[16a] receiving, by the first computer system, over a network, a copy of a first file from a second computer system associated with a user, wherein the copy of the first file is automatically received from the second computer system responsive to the user modifying a content of the first file stored on the second computer system, the copy of the first file being a version of the first file that is generated from the user modifying the content of the first file;

[16b] storing, by the first computer system, the copy of the first file on the first computer system;

[16c] receiving, by the first computer system, from the second computer system, first metadata associated with the version of the first file that is generated from the user modifying the content of the first file, the first metadata being

assigned a first priority greater than a second priority assigned to the copy of the first file;

[16d] automatically transferring, by the first computer system, based on the first priority being greater than the second priority, the first metadata to a third computer system associated with the user such that the first metadata is transferred to the third computer system prior to the copy of the first file being transferred to the third computer system; and

[16e] automatically transferring, by the first computer system, over a network, the copy of the first file to the third computer system associated with the user to replace an older version of the first file stored on the third computer system, responsive to receiving the copy of the first file from the second computer system.

Claim 17: The one or more non-transitory machine-readable media of claim 16, wherein at least one of the first computer system or the second computer system comprises a priority assignment configuration to assign greater priority to metadata associated with files than priority assigned to the files such that at least one of the first computer system or the second computer system assigns the first priority to the first metadata and the second priority to the copy of the first file based on the priority assignment configuration.

CERTIFICATE OF COMPLIANCE WITH WORD COUNT

Pursuant to 37 C.F.R. § 42.24(d), I certify that this petition complies with the type-volume limits of 37 C.F.R. § 42.24(a)(1)(i) because it contains 13,607 words, according to the word-processing system used to prepare this petition, excluding the parts of this petition that are exempted by 37 C.F.R. § 42.24(a) (including the table of contents, a table of authorities, mandatory notices, a certificate of service or this certificate word count, appendix of exhibits, and claim listings).

DATED: January 5, 2023

COOLEY LLP
ATTN: Patent Docketing
1299 Pennsylvania Avenue NW
Suite 700
Washington, D.C. 20004
Tel: (202) 842-7889
Fax: (202) 842-7899

/Heidi L. Keefe/
Heidi Keefe
(Reg. No. 40,673)
COOLEY LLP
ATTN: Patent Group
1299 Pennsylvania Ave., NW
Suite 700
Washington, D.C. 20004
Tel: (650) 843-5001
Fax: (650) 849-7400
hkeefe@cooley.com

Counsel for Petitioners

CERTIFICATE OF SERVICE

I hereby certify, pursuant to 37 C.F.R. Sections 42.6 and 42.105, that a complete copy of the attached **PETITION FOR INTER PARTES REVIEW OF U.S. PATENT NO. 11,003,622**, including all exhibits (**Nos. 1001-1028**) and related documents, are being served via Federal Express on the 5th day of January, 2023, the same day as the filing of the above-identified documents in the United States Patent and Trademark Office/Patent Trial and Appeal Board, upon the PO by serving the correspondence address of record with the USPTO as follows:

John Bird
Andrew Taska
Sughrue Mion, PLLC
2000 Pennsylvania Ave., N.W.
Suite 9000
Washington, D.C. 20006

The same documents are also being sent via Federal Express upon counsel of record for the PO in the litigations pending before the U.S. District Court for the Western District of Texas entitled *Topia Technology, Inc. v. Box, Inc.*, Case No. 6:21-cv-01372-ADA (W.D. Tex.), and *Topia Technology, Inc. v. Dropbox, Inc.*, Case No. 6:21-cv-01373-ADA (W.D. Tex.), as follows:

Chidambara Iyer, Esquire
Mark Boland, Esquire
L. Roman Rachuba, Esquire
SUGHRUE MION, PLLC
2000 Pennsylvania Ave., N.W.
Suite 9000
Washington, D.C. 20006

Raymond W. Mort, III, Esquire
THE MORT LAW FIRM, PLLC
100 Congress Ave, Suite 2000
Austin, TX 78701

DATED: January 5, 2023 /Heidi L. Keefe/
Heidi Keefe
(Reg. No. 40,673)