

Petition for *Inter Partes* Review of
U.S. Patent No. 10,067,942

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. IPR2023-00433

U.S. Patent No. 10,067,942

Filing Date: September 21, 2015

Issue Date: September 4, 2018

Title: Architecture for Management of Digital Files Across Distributed Network

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 10,067,942**

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 '942 Patent.....	5
VI. Level of Ordinary Skill in the Art	6
VII. Claim Construction.....	6
VIII. The Challenged Claims are Unpatentable	6
A. Earliest Effective Filing Date.....	6
B. Overview of Grounds Including Prior Art Qualification	7
C. Summary of Prior Art.....	8
B. Ground 1: Obviousness of Challenged Claims over Sigurdsson in combination with Shappell.....	12

Table of Contents

(continued)

	Page
C. Ground 2: Obviousness of Challenged Claims over Brown in combination with Hesselink.....	40
IX. Conclusion	65

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, <i>Microsoft Outlook Version 2002 Inside Out</i> (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	<i>Topia Technology, Inc. v. Box, Inc.</i> , Case No. 6:21-cv-1372-ADA, D.I. 60 (Joint Claim Construction Chart)
1028	Declaration of Dr. Ravin Balakrishnan

Petition for *Inter Partes* Review of
U.S. Patent No. 10,067,942

I. INTRODUCTION

Petitioners Box, Inc., and Dropbox, Inc., respectfully petition for *Inter Partes* Review (“IPR”) of claims 1, 3, 4, 10, 12 and 13 (“Challenged Claims”) of U.S. Patent No. 10,067,942 (“’942”).

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. The other defendants in the pending litigations, Vistra Corp., SailPoint Technologies Holdings, Inc., Clear Channel Outdoor Holdings, Inc., and Egnyte, 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 ’942 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; 10,642,787; 10,289,607; 10,754,823; and 11,003,622 (“Related

Petition for *Inter Partes* Review of
U.S. Patent No. 10,067,942

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 the ’942 was instituted on November 17, 2022. IPR2022-00782, Decision Granting Institution, Paper 17 (P.T.A.B. Nov. 17, 2022) (P.T.A.B. Nov. 17, 2022) (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 ’942 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 <i>pro hac vice</i> 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 <i>pro hac vice</i> to be requested) lmead@cooley.com COOLEY LLP

Petition for *Inter Partes* Review of
U.S. Patent No. 10,067,942

LEAD COUNSEL	BACK-UP COUNSEL
	ATTN: Patent Group 1299 Pennsylvania Avenue NW 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 '942, 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 and electronically 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 six claims of the '942. A payment of \$41,500 is being 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 '942 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, 4, 10, 12 and 13	Obvious by <u>Sigurdsson</u> in view of <u>Shappell</u>
2	1, 3, 4, 10, 12 and 13	Obvious by <u>Brown</u> in view of <u>Hesselink</u>

Submitted with this Petition is the Declaration of Dr. Ravin Balakrishnan, a qualified technical expert (EX1028) ("Balakrishnan").¹ (Balakrishnan, ¶¶1-12.)

¹ Given the significant overlap in claimed subject matter of the '942 and Related Patents, Dr. Balakrishnan's Declaration addresses all Related Patents in addition to the '942. This Petition cites only the portions related to the '942.

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): The presented references were not cited during prosecution and are not cumulative of cited references. (Balakrishnan, ¶¶51, 56.)

V. OVERVIEW OF '942 PATENT

A. Claim Listing

Appendix A lists the Challenged Claims.

B. Specification

The '942 relates to file transfer and synchronization between a user's multiple electronic devices. ('942, 1:18-20.) Files modified on one user device may be temporarily stored on a server before being forwarded to another user device to maintain consistency between versions of the files stored on each user device. (*Id.*, 8:16-25.) *See also* Unified ID at 7-9 (summarizing specification). The patent

Petition for *Inter Partes* Review of
U.S. Patent No. 10,067,942

contemplates automatically transferring copies of modified files between a user's devices for file synchronization. ('942, 3:50-4:2.)

VI. LEVEL OF ORDINARY SKILL IN THE ART

As discussed below, the claims' earliest effective filing date is November 10, 2008. A person of ordinary skill in the art ("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. (Balakrishnan, ¶¶22-27.) In litigation, PO proposes that no terms need construction; Petitioners propose a construction for 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 '942 claims priority to a 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

Petition for *Inter Partes* Review of
U.S. Patent No. 10,067,942

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 (citation 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]-[1i] and [10a]-[10h],² so the claims’ effective priority is no earlier than 11/10/2008. (Balakrishnan, ¶¶65, 103-109.) 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:

<u>Reference</u>	<u>Filed</u>	<u>Published/Issued</u>	<u>§102 (pre-AIA)</u>
Sigurdsson	1/25/2006	7/26/2007	§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)

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

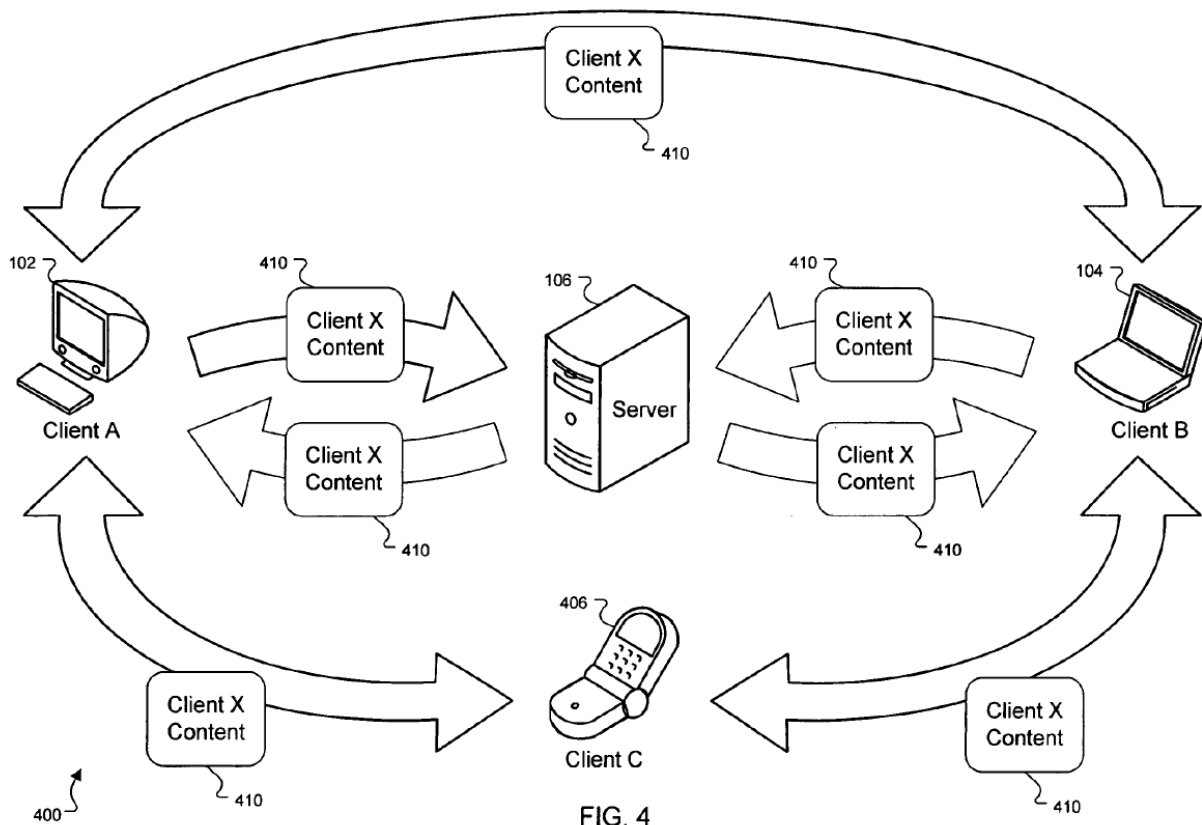
Ground 1 relies primarily on **Sigurdsson**, which discloses the claims' core synchronization features, as the Examiner found in a related patent prosecution (EX1012 at 118-140), in combination with **Shappell**.

In litigation, PO has claimed a priority date would allegedly pre-date 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**, which discloses the claims' core synchronization features, in combination with **Hesselink**.

C. Summary of Prior Art

1. Sigurdsson [EX1016]

Sigurdsson is strikingly similar to the challenged patents, discussing substantially similar needs and proposed solutions. (Sigurdsson, ¶¶2-14.) Sigurdsson describes synchronizing files across a user's multiple client devices. (Sigurdsson, ¶¶26-27; Balakrishnan, ¶¶118-120.) Figure 4 shows the relevant system:



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, 112-13, 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, ¶¶121-122.) “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; Balakrishnan, ¶119.) Content sent from clients to Server 106 may include copies of files and metadata. (Sigurdsson, ¶54; Balakrishnan, ¶124.) Using

a “priority algorithm,” different items may be prioritized differently for synchronization, based on factors such as “size, type, or age.” (Sigurdsson, ¶58; Balakrishnan, ¶¶125-26.) As a result, metadata may be sent to Server 106 prior to associated files, such as when the files are large. (Sigurdsson, ¶72; Balakrishnan, ¶126.) 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, ¶72; Balakrishnan, ¶126.)

2. Shappell [EX1020]

Shappell teaches “shared spaces” where files are shared and synchronized among group members. (Shappell, ¶76; Balakrishnan, ¶141.) 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, ¶¶76, 78-79; Balakrishnan, ¶141.) 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; Balakrishnan, ¶144.)

3. Brown [EX1025]

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

Petition for *Inter Partes* Review of
U.S. Patent No. 10,067,942
account.”); Balakrishnan, ¶561.)

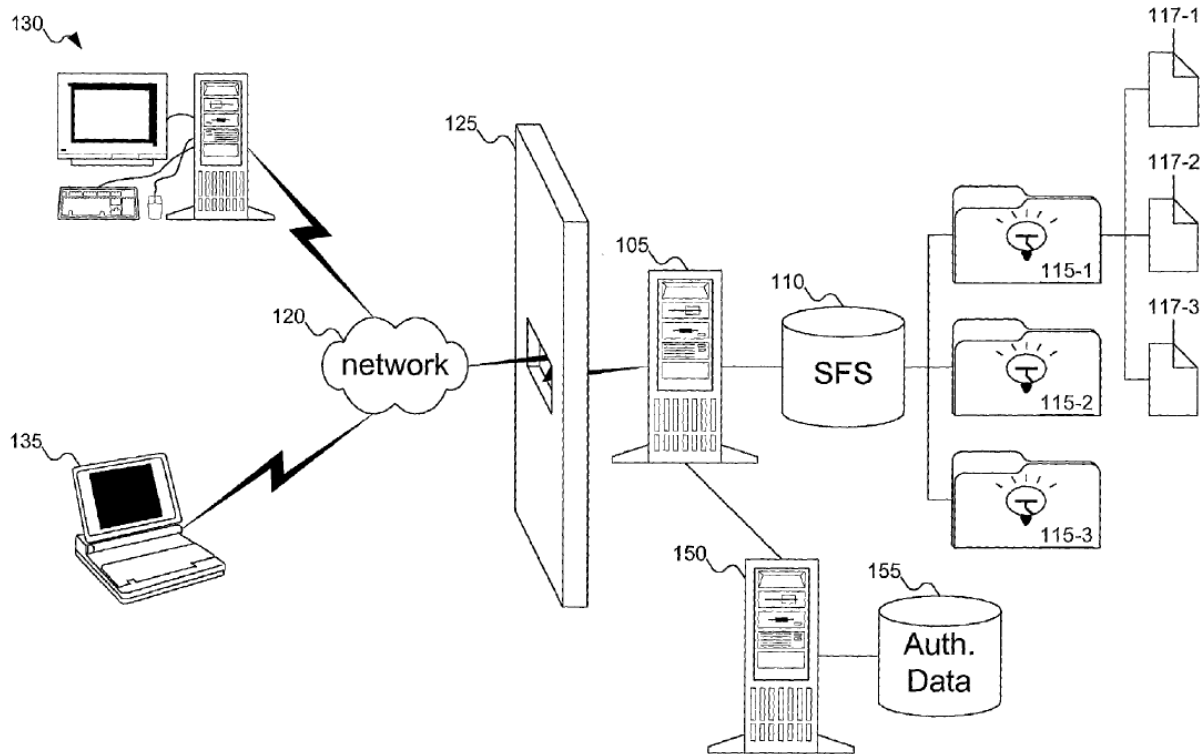


FIG. 1

(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; Balakrishnan, ¶561.) 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; Balakrishnan, ¶¶561-566.)

4. Hesselink [EX1026]

Like Brown, Hesselink discloses a synchronization system involving multiple

user computers, such as home, work, and laptop computers. (Hesselink, ¶210; Balakrishnan, ¶569.) 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; Balakrishnan, ¶569.) 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; Balakrishnan, ¶570). Hesselink’s system “automatically and repeatedly” checks whether synchronizing devices are in communication prior to initiating synchronization. (Hesselink, ¶182, 189; Balakrishnan, ¶571.)

B. Ground 1: Obviousness of Challenged Claims over Sigurdsson in combination with Shappell.

1. Claim 1.

(a) [1pre]³

Sigurdsson discloses a content sharing and synchronization “**system**” comprising clients A-C (**102, 104, 406**) and server 106 as shown in Figure 4:

³ See **Appendix A** for bracketed elements.

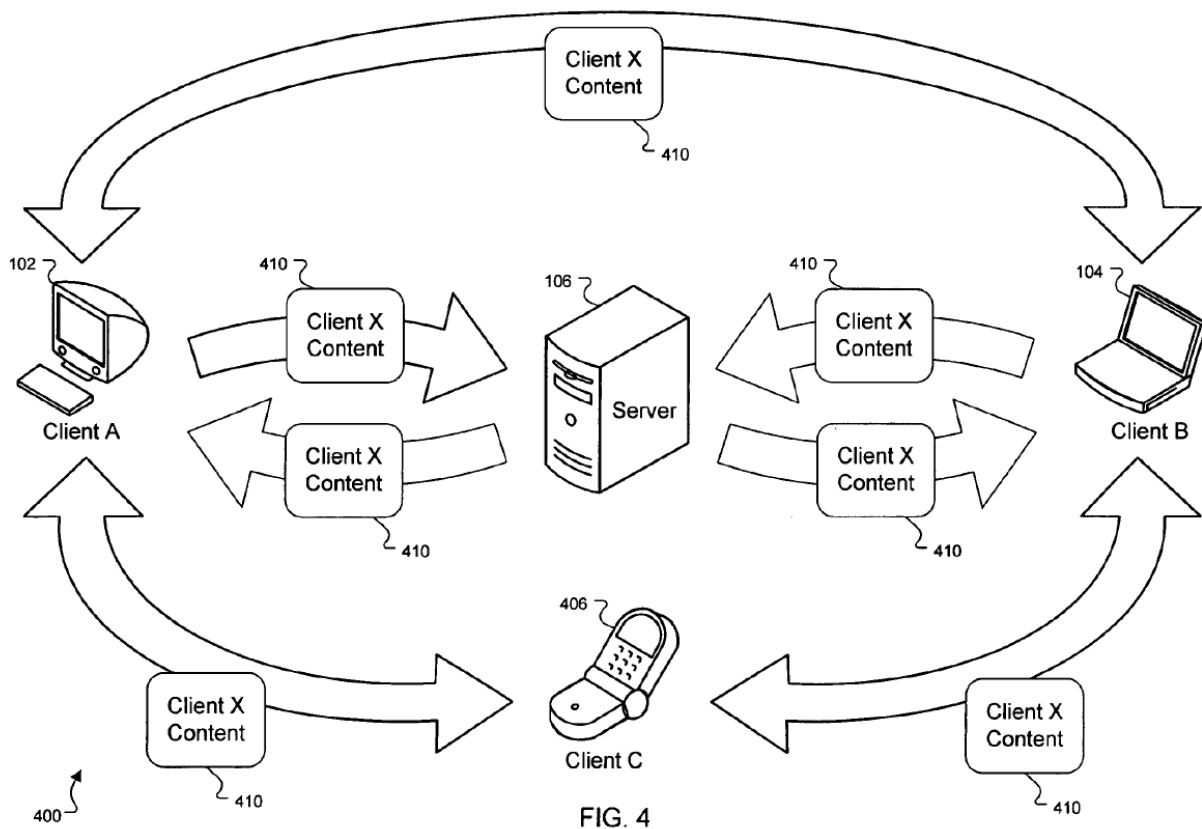


FIG. 4

(Sigurdsson, Fig. 4, ¶¶45-51; Balakrishnan, ¶¶146-149, 413.)

(b) [1a]

“Server 106” in Figure 4 (and as described in connection with Figure 9 and elsewhere in Sigurdsson) is “a **first electronic device**” that may be “**configured.**” (Balakrishnan, ¶¶414-416.) For example, Sigurdsson discloses that “System 900” depicted in Figure 9 “may be included in . . . Server 106.” (Sigurdsson, ¶104; Balakrishnan, ¶415.) System 900 includes “a **Processor 910**, a **Memory 920**, a **Storage Device 930**” and “[t]he Processor 910 is capable of processing instructions for execution within the System 900.” (Sigurdsson, ¶105; Balakrishnan, ¶415.) Server 106 is therefore “**configured**” to execute, using “Processor 910,”

“instructions stored in the Memory 920 or on the Storage Device 930[.]”

(Sigurdsson, ¶105; *see also id.*, ¶¶109-110; Balakrishnan, ¶415.)

Sigurdsson discloses that Server 106 may be **“associated with a user.”** (Balakrishnan, ¶416.) For example, Server 106 may be associated with the user because the Server can store and authenticate a user ID for each user. (Sigurdsson, ¶49 (“The Authentication Manager 334 can check the credentials of a client when the client connects to the Server 106. If a client supplies invalid credentials, the Authentication Manager can transmit an error message to the client.”); *see also id.*, ¶27 (“The **Server 106** can provide a temporary storage location, or ‘drop box’, for **a user’s** files”); Balakrishnan, ¶416.)

(c) [1b]

Server 106 includes a “Synchronization Manager 332,” which a POSA would understand to be a **“first application.”**⁴ (Sigurdsson, ¶48; Balakrishnan, ¶¶417-

⁴ It would have been obvious to combine one or more functionalities performed by the **“first application”** in claim 1 into a single application. (Balakrishnan, ¶418 n.16.) For example, it would have been obvious to include the functionality for sending and receiving data between clients and the server and the functionality for managing storage on the server in a single application on the server, as this would have been within the knowledge of a POSA. (*Id.*)

428.) As shown in Sigurdsson's Figure 3 and accompanying text, such as in paragraph 48, Server 106 contains only two applications (the Synchronization Manager 332 and Authentication Manager 334). (Sigurdsson, ¶48; Balakrishnan, ¶418.) The Authentication Manager 334 would not be understood by a POSA to perform the relevant claimed functions, since Authentication Manager 334 is described only as handling authentication/security functions. (Sigurdsson, ¶49; Balakrishnan, ¶418.) Thus, it would have been obvious to a POSA that Synchronization Manager 332 is used to perform the relevant claimed functions associated with the "**first application**," such as sending, receiving, and storing data. (Sigurdsson, ¶¶48, 112; Balakrishnan, ¶418.)

Sigurdsson discloses Client A ("**second electronic device**") and Client C ("**third electronic device**"), which may be in communication with Server 106 ("**first electronic device**"), as shown in Figure 4. (Sigurdsson, ¶51 ("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, ¶419.) Each of Sigurdsson's client devices are described as being "**associated with a user**." (Sigurdsson, ¶31; Balakrishnan, ¶428.) For example, Client A is described as a "user's home computer," Client B is described as a "user's work computer," and Client C is

Petition for *Inter Partes* Review of
U.S. Patent No. 10,067,942

depicted in Figure 4 as a user's mobile device. (Sigurdsson, Fig. 4, ¶¶2, 27-29, 55, 58 (Clients A, B, and C are associated with the same user); Balakrishnan, ¶428.)

A POSA would have appreciated and recognized Sigurdsson's teachings of the obvious advantages of a client/server arrangement. (Balakrishnan, ¶427.) 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, ¶427.) A server may also have more available bandwidth relative to clients for transferring data. (Sigurdsson, ¶79; Balakrishnan, ¶427.) Sigurdsson makes clear the peer-to-peer and client/server pathways between the user's multiple client devices are interchangeable, rendering it obvious to synchronize through the server. (Sigurdsson, ¶51; Balakrishnan, ¶427.) For example, Sigurdsson explains that each client may include an "HTTPS Client/server" that may interchangeably be used for synchronization with Server 106 (Sigurdsson, ¶54) *or* synchronization with other clients (*id.*, ¶47). (Balakrishnan, ¶427.) Even for embodiments in which clients may communicate peer-to-peer, Sigurdsson teaches that they can additionally use Server 106 for content sharing. (Sigurdsson, ¶¶51, 79 ("Direct content sharing among peer clients can make use of available resource capacities at the clients, thus reducing the bandwidth traffic at the

Petition for *Inter Partes* Review of
U.S. Patent No. 10,067,942

Server 106. *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.”); Balakrishnan, ¶427.)

Server 106 (“**first electronic device**”) is configured to “**receive . . . a copy of a modified first electronic file**” in the form of, for example, a “very large” “maintenance manual.” (Sigurdsson, ¶58; Balakrishnan, ¶¶418-421.) The maintenance manual is received from a “**second application**” executable on Client A (“**second electronic device**”) in the form of an HTTPS client/server: “The *Client A 102* can include both data and *applications*. . . . The *applications* include an Event Listener 322, a File to Text/HTML Converter 324, and *a hypertext transfer protocol secure (HTTPS) Client/server 326*.” (Sigurdsson, ¶47; Balakrishnan, ¶420.) A POSA would understand that the HTTPS Client/server application transfers content from Client A to the server via, for example, an HTTP “post” command. (*E.g.*, Sigurdsson, ¶54 (“For example, referring to FIG. 4, 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, ¶420.)

Server 106 may “**automatically receive[]**” “**the modified first electronic file copy**” (maintenance manual) “**from the second application**” (HTTPS Client/server on Client A) “**responsive to the user modifying a content of the first electronic file.**” (Sigurdsson, Fig. 4, ¶¶58, 61; Balakrishnan, ¶421.) For example, Client A

may send a file to Server 106 “***when content is changed***” in the file. (Sigurdsson, ¶84 (“The Client A 102 posts Client A Content 108 to the Server 106 ***when content is changed*** on the Client A 102. . . . [P]osting can occur as soon as an event is generated by an action, such as saving a file. For example, a client can maintain an open connection with a server and post as soon as an event is generated.”); *see also id.*, ¶¶32-33, 45, 47, 62, 63, 68, 51 (Figure 4 components can share content using method 200 of Figure 2 implementing Figure 1 client/server system); Balakrishnan, ¶¶422-423.)

The content transferred to the server may include a full copy of the file modified by the user. (Sigurdsson, ¶65 (“In step 606, the content originating on the first client is posted [to the server]. ***The content can include a copy of the document that the user saves***”); *see also id.*, ¶¶8, 28, 103; Balakrishnan, ¶424.)

As a result of Client A posting content to the server, “[t]he Client A Content 108 is received by and stored in the Server 106.” (Sigurdsson, ¶85; *see also id.*, ¶48 (“The ***Server 106 can contain a Storage 330***, a Synchronization Manager 332, and an Authentication Manager 334. The ***Storage 330*** can contain the Content A, and Content B through Content N 336 that has been received from the user’s various clients.”); *id.*, ¶¶74, 83, 85, 104, 110; Balakrishnan, ¶425.)

Although some embodiments refer to peer-to-peer sending of files (e.g., from Client A to Client B), Sigurdsson describes that files may be sent directly to the

Petition for *Inter Partes* Review of
U.S. Patent No. 10,067,942

server system by a client, including in the embodiments described in Figures 4-7.

(Balakrishnan, ¶426.) Client A may “*send[] the data directly to the server*, as shown by the dashed line pointing from the step 606 to the step 610. For example, referring to FIG. 4, the *Client X Content 410 can be sent directly to the Server 106 . . .*”

(Sigurdsson, ¶¶68, 84-85, 51; Balakrishnan, ¶426.)

(d) [1c]

It would have been obvious to use Server 106 to “**determine whether the first electronic device**” (Server 106) “**is in communication with a third electronic device**” (Client C). (Balakrishnan, ¶¶429-431.) Sigurdsson discloses, in connection with the embodiment of Figure 6, 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, ¶430.) In this embodiment, “*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, ¶430.) A notification sent to Server 106 from Client C when Client C

Petition for *Inter Partes* Review of
U.S. Patent No. 10,067,942

comes online allows Server 106 to determine that Server 106 is in communication with Client C. (Sigurdsson, ¶¶66, 73 (describing embodiment in which Client C is offline and comes back online); Balakrishnan, ¶430.)

Sigurdsson's clients operate similarly and are interchangeable, so it would have been obvious to apply the functionality in this limitation to Clients A, B, or C. (Balakrishnan, ¶431.) For example, Sigurdsson explains that "Clients A and B 102, 104 can contain similar components and can accept and transmit content in a similar manner. For example, both clients can include Files 346, HTML Files 348, an Index 350, an Event Listener 352, a File to Text/HTML Converter 354 and an HTTPS Client/server 356." (Sigurdsson, ¶50; Balakrishnan, ¶431.) Sigurdsson further explains that Client C operates similarly to Clients A and B, and it therefore would have been obvious that Client C could contain similar components to Clients A and B. (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 communicat[ion] with the Server 106 in a manner similar to the method used by the Clients A or B.*"); Balakrishnan, ¶431; *B/E Aerospace v. C&D Zodiac*, 962 F.3d 1373, 1379-80

(Fed. Cir. 2020) (affirming obviousness of “predictable” variation).) Because Clients A, B, and C are described as operating similarly, they all could post/push files to Server 106 in the same way using the same components. (*See* Sigurdsson, ¶51, Fig. 3 (depicting both Client A and Client B as having identical event listeners and HTTPS client/servers); Balakrishnan, ¶431.) In addition, the functionality related to transferring data from Client A to Server 106 and then from Server 106 to other clients is described generically as applying to any combination of clients and the server. (Sigurdsson, ¶68 (“For example, referring to FIG. 4, the Client X Content 410 can be sent directly to the Server 106 The Server 106 can then transmit the content to the remaining client peers, such as the Clients B and C.”); Balakrishnan, ¶431.) Thus, the functionality for synchronizing files related to any particular client described in Sigurdsson could apply to any other client as well (the clients are fungible), and the server may detect whether Clients A, B, or C are online or offline and in communication or out of communication with the server. (Balakrishnan, ¶431.)

(e) [1d]

Sigurdsson discloses and renders obvious to “**automatically send, via the first application**” (Synchronization Manager 332) “**the modified first electronic file copy**” (updated copy of the modified maintenance manual) to a “**third application**” (HTTPS Client/server) “**at the third electronic device**” (Client C).

(Balakrishnan, ¶¶432-446.) As discussed above for claim [1c], Client C may include an HTTPS Client/server like those of Clients A and B for sending and receiving files/content to and from the server. (Balakrishnan, ¶433.) In connection with the embodiment in Figure 6, Sigurdsson explains that Server 106 may send content, such as the modified maintenance manual received from Client A, to another client, such as Client C (via its HTTPS Client/server), to keep Client C up-to-date with the latest file changes from other clients. (Sigurdsson, ¶80 (“In step 626, the server sends the missing content to the third client. For example, referring to FIG. 4, the **Server 106 sends the Client X Content 410 to the Client C 406.**”); *see also id.*, ¶51 (Client C may communicate bidirectionally with Server 106); Balakrishnan, ¶433.)

It would have been obvious to practice this limitation in view of additional examples in Sigurdsson, which teach and suggest receiving Client A content at Server 106 and automatically transferring the content to Client B, C, or any other client. (Sigurdsson, ¶68 (“For example, referring to FIG. 4, the Client X Content 410 can be sent directly to the Server 106 **The Server 106 can then transmit the content to the remaining client peers, such as the Clients B and C.**”); Balakrishnan, ¶436.) In one embodiment, Sigurdsson describes an automated process for (1) detecting when Client C comes back online after being offline, (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. (Sigurdsson, ¶¶69-80; Balakrishnan, ¶436.) As another example, it would have been obvious in view of the embodiment in Figure 5 showing synchronization between Clients A and B to instead use Clients A and C, given the interchangeability of Sigurdsson's clients, as discussed above for claim [1c]. (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."); *see also id.*, ¶31 ("When the Client A 102 (e.g., the user's home computer) connects to the Server 106, Client B Content 112 originating from the Client B 104 is automatically copied to the Client A 102, as indicated by arrow 116. . . . Similarly, whenever the Client B 104 (e.g., the user's work computer) connects to the Server 106, any missing Client A Content 108 is copied automatically to the Client B 104, as indicated by arrow 118."); ¶84 ("[A] client can maintain an open connection with a server and post as soon as an event is generated."); Balakrishnan, ¶¶438, 442.)

Sigurdsson also discloses and renders obvious that the first electronic file copy is sent **"responsive to the determination that the first electronic device is in communication with the third electronic device."** (Balakrishnan, ¶441.) As discussed above in connection with claim [1c], Sigurdsson discloses and renders obvious determining whether the server is in communication with Client C. (Balakrishnan, ¶¶429-431, 434.) Only after this determination is made, and only in response to this determination, is the file sent to Client C. (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. . . . 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, ¶434.)

A POSA also would have been motivated to use these teachings and Server 106 to “**automatically**” send the modified first electronic file copy “**responsive to receiving the modified first electronic file copy from the second electronic device.**” (Balakrishnan, ¶435.) Server 106 includes a “Synchronization Manager” that “synchronize[s] content stored on and shared between” different clients (e.g., “Client A” and “Client C”). (See Sigurdsson, ¶48 (“The Synchronization Manager 332 can use the Time Stamps 338 to synchronize content stored on and shared between the Client A 102 and the Client B 104.”); Balakrishnan, ¶435.) A POSA would understand that “synchronization” in this context means “version comparisons of copies of . . . files to ensure they contain the same data.” (EX1021, Microsoft Computer Dictionary, Fifth Edition (2002), 506 (definition of “synchronization”); see also *id.*, (“In handheld computing, the process of updating or backing up the data on a handheld computer to the linked software applications on a desktop computer. Data changes made on the desktop computer may also be copied to the handheld during synchronization.”); Balakrishnan, ¶435.) Thus, a

POSA would have understood and found it obvious that Sigurdsson's reference to "synchroniz[ing]" content stored on and shared between clients would have encompassed using the Synchronization Manager on Server 106 to automatically transfer a modified first electronic file copy received from Client A to Client C. (Sigurdsson, ¶¶69-80 (sending content received from Client A to Client C via Server 106 when Client C comes back online); Balakrishnan, ¶435.)

It would have been obvious in view of Sigurdsson that the "**first electronic file**" could be received and stored on the "**third electronic device.**" (Sigurdsson, ¶¶81, 104; Balakrishnan, ¶455.) In addition, it would have been obvious that, instead of implementing Client C as a mobile device as shown in Figure 4 (which may have had limited memory in November 2008), Client C could have been implemented as a computer with memory large enough to store a large file, such as an updated maintenance manual, for the reasons discussed above in connection with claim [1c] regarding the obviousness of implementing functionality associated with Client A using a similarly configured Client C. (Sigurdsson, ¶¶45, 50-51; Balakrishnan, ¶455; *B/E Aerospace*, 962 F.3d at 1379-80 (affirming obviousness of "predictable" variation).)

As discussed above for claim [1b], the content transferred to the server by Client A may include a full copy of the user's modified file. (Sigurdsson, ¶¶8, 28, 54, 65; Balakrishnan, ¶¶417-428.) For example, it would have been obvious that the

same “Client X Content” received from Client A, including a full copy of the modified file, could have been sent from the server to Client C when Client C comes back online after being offline. (Sigurdsson, ¶68 (“Client X Content” may be “sent directly to the Server 106” from an originating client, such as Client A); *id.*, ¶80 (“[T]he Server 106 sends the Client X Content 410 to the Client C 406.”); Balakrishnan, ¶436.)

(f) [1e]

It would have been obvious to implement the functionality performed by Client A in claim [1b] but instead using Client C (“**third electronic device**”) and an HTTPS Client/server on Client C (“**third application**”) to automatically transfer a “**copy of a modified second electronic file**” (e.g., “a small office memo”) to the server when the file is modified. (Sigurdsson, ¶58 (“[T]he user may go to work and *update a small office memo on the Client C* 406.”); Balakrishnan, ¶¶447-448.) As discussed above for claim [1b], Client C is “**associated with the user.**” (Balakrishnan, ¶¶417-428.) As discussed above for claim [1c], Sigurdsson describes Client C as operating similarly to Client A, and it would have been obvious to include the same components in Clients A and C to implement the functionality recited in claim [1b] using Client C to automatically transfer a file to the server. (Sigurdsson, ¶¶50-51, 68, Fig. 3; Balakrishnan, ¶¶429-431, 448.)

(g) [1f]

For the reasons discussed above in connection with claim [1c], Sigurdsson's clients operate similarly and are interchangeable, so it would have been obvious to implement the functionality described in claim [1c] for determining whether the first electronic device is in communication with the third electronic device but instead for **"determin[ing] whether the first electronic device is in communication with the second electronic device."** (Balakrishnan, ¶¶449-450.) For example, Sigurdsson discloses determining whether "Client B" is offline instead of Client C (and, of course, it would have been obvious to also detect whether the server is in communication with Client A ("**second electronic device**"), given that Clients A and B include similar components and perform similar functions). (Sigurdsson, ¶50 ("Clients A and B 102, 104 can contain similar components and can accept and transmit content in a similar manner. For example, both clients can include Files 346, HTML Files 348, an Index 350, an Event Listener 352, a File to Text/HTML Converter 354 and an HTTPS Client/server 356."); Balakrishnan, ¶450; *B/E Aerospace*, 962 F.3d at 1379-80.) Thus, for the same reasons above for claim [1c], Sigurdsson discloses that its server may detect whether Clients A, B, or C are online or offline and in communication or out of communication with the server. (Balakrishnan, ¶450.)

(h) [1g]

For the reasons discussed above for claim [1c], Sigurdsson's clients operate similarly and are interchangeable. (Balakrishnan, ¶¶451-452.) Thus, for the reasons discussed above in connection with claims [1c] and [1d], it would have been obvious to automatically transfer the modified **"second electronic file copy"** (small office memo) from the server, via the server's **"first application,"** to the **"second application"** (HTTPS Client/server at Client A) **"responsive to the determination that the first electronic device"** (Server 106) **"is in communication with the second electronic device"** (Client A) **"and responsive to receiving the modified second electronic file copy from the third electronic device"** (Client C). (Balakrishnan, ¶452.)

(i) [1h]

This limitation would have been obvious in view of Sigurdsson in combination with Shappell. For example, it would have been obvious in view of Sigurdsson in combination with Shappell that, **"responsive to sending the modified first electronic file copy to the third electronic device, an older version of the first electronic file stored on the third electronic device is automatically caused to be replaced with the modified first electronic file copy such that the modified first electronic file copy is stored on the third electronic device in lieu of the older version of the first electronic file."** (Balakrishnan, ¶¶453-462.) Like

Sigurdsson, Shappell discloses automatically transferring a modified file to another client after receiving the modified file from an originating client. (Balakrishnan, ¶¶461-462.) Shappell also teaches a user-selectable “File Replication” setting by which 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, ¶461.) When “automatic replication” is enabled, Shappell teaches automatically replacing an out-of-date file with an updated version on a receiving computer. (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, ¶461.)

A POSA would have been motivated to apply these concepts from Shappell to Sigurdsson’s systems and methods. (Balakrishnan, ¶¶459-462.) Sigurdsson discloses concerns about the limited memory on Server 106 and the need to manage the amount of data stored on Server 106. (Balakrishnan, ¶440.) For example, Synchronization Manager 332 includes “a Space Quota 342 and a Throughput

Threshold 344” that are used to control the storage of data to the server. (Sigurdsson, ¶48; Balakrishnan, ¶440.) A POSA would have been motivated in view of these concerns to automatically transfer an incoming, updated file from Server 106 to any other client device that may need the file as quickly as possible and to replace any older version of the file stored on Server 106 with the newer version to avoid unnecessary use of Server 106’s memory. (Balakrishnan, ¶440.)

In addition, 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 on each respective version. (Balakrishnan, ¶459.) For example, Sigurdsson expressly references a client’s work computer and home computer as two exemplary “clients,” and describes example use cases such as a user editing a Word document that is synchronized between the user’s home and work computers: “For example, a user may edit a word document file at his home computer. When the user saves the document, the document’s content is automatically extracted and transmitted to the user’s computer at work. When the user searches for the modified document later at his work computer, the content is presented to the user even though the word document was not previously stored on the user’s work computer.” (Sigurdsson, ¶¶26-28 (also describing Server 106 as optional “drop box” “conduit” between client devices); Balakrishnan, ¶459.)

A POSA would have found it obvious that in this type of use case, after the copy of the Word document (which may be called “Version 1”) is transmitted from the user’s home computer to the user’s work computer, the user may continue to edit that document on the work computer while the user is at work. (Balakrishnan, ¶460.) Based on Sigurdsson’s bidirectional file synchronization architecture as illustrated in Figure 4, a copy of the updated, edited file at the work computer (call it “Version 2”) would then be transmitted from the work computer, through the Server, and back to the user’s home computer. (*Id.*) However, Sigurdsson does not provide details about what would happen at the user’s home computer in that circumstance where “Version 2” of the file is received at the home computer that previously contained “Version 1” of the same file. (*Id.*) A POSA would have been motivated in view of Sigurdsson’s concerns with memory management to look for solutions to replace old versions of files with new ones as described in this limitation. (*Id.*) A POSA would have found it obvious that one possibility would be for the updated modified file (“Version 2”) to replace the outdated older version of the file (“Version 1”) on the home computer. (*Id.*) A POSA would have found it obvious to replace the older version of the file for reasons including the obvious benefit of preventing copies of outdated versions from endlessly accumulating in storage on the user’s computers. (Balakrishnan, ¶¶460, 462.)

Motivation to Combine Sigurdsson with Shappell

It would have been obvious to combine Sigurdsson with Shappell's teachings, predictably resulting in Sigurdsson's system where a modified file received from an originating client is automatically forwarded to a receiving client after receipt of the file at Server 106, to replace an older version of the file stored at the receiving client.

Sigurdsson and Shappell are analogous references in the same field as the '942, directed to synchronizing files. (Balakrishnan, ¶¶182, 461-462.) 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 disclosed systems, automatically transferring a modified file received at a server from an originating client to a receiving client to replace an older version of the file stored at the receiving client. (Balakrishnan, ¶¶183, 462.)

A POSA also would have been motivated to consider Shappell given its origination 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.)

A POSA would have reasonably expected success in implementing this modified Sigurdsson system, as Sigurdsson already discloses automated transfer of a file received from one client to another client via a server. (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 the type of network architecture as between peer-to-peer or client-server. (*Id.*) 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, in its exemplary Figure 2 embodiment, "[e]ach user device will *typically* be used by or associated with a different user" (Shappell, ¶40), given the compelling applicability of Shappell's relevant teaching to improve Sigurdsson's system. (Balakrishnan, ¶189.)

(j) [1i]

As noted above for claim [1c], Sigurdsson explains that its clients operate similarly and are interchangeable. (Balakrishnan, ¶¶463-464.) Thus, for the reasons discussed above in connection with element [1h], it would have been obvious to implement the functionality described in element [1h] but with respect to the "second electronic file copy" instead of the first electronic file copy and the "second electronic device" instead of the third electronic device. (Balakrishnan, ¶464.)

2. Claim 3.

Sigurdsson discloses that Server 106 is “**configured to**” “**store**” “**via the first application**” (Synchronization Manager 332) “**the modified first electronic file copy**” (modified copy of the maintenance manual) “**to a memory device associated with the first electronic device**” (e.g., Storage 330). (Sigurdsson, ¶48 (“*The Server 106 can contain a Storage 330, a Synchronization Manager 332, and an Authentication Manager 334. The Storage 330 can contain the Content A, and Content B through Content N 336 that has been received from the user’s various clients.*”); *id.*, ¶74 (“the content may be stored at the server”); Balakrishnan, ¶¶465-467.) Sigurdsson further discloses: “The Server 106 serves as a *temporary storage location* that can facilitate synchronization between the user’s clients.” (Sigurdsson, ¶83; *see also id.*, ¶¶85, 104, 110; Balakrishnan, ¶466.) As discussed above, Server 106’s “Synchronization Manager 332” is an exemplary “**first application**,” which includes “a Space Quota 342 and a Throughput Threshold 344” that are used to control the storage of data to the server. (Sigurdsson, ¶48; Balakrishnan, ¶467.) In view of these disclosures, a POSA would have found it obvious that Synchronization Manager 332 manages not only the sending and receiving of files to and from clients but also the storage of such files on Server 106. (Balakrishnan, ¶467.)

3. Claim 4.

As discussed above in connection with claim 3, Sigurdsson discloses that **“the modified first electronic file copy is stored on the memory device associated with the first electronic device.”** (Balakrishnan, ¶¶468-480.) 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 computer, work computer, and Server 106 and the issue of synchronizing all three devices to make sure they are updated with the most recent updates on each respective version. (Balakrishnan, ¶470.) For example, as discussed above for claim [1h], Sigurdsson references a client’s work computer and home computer as two exemplary “clients,” and describes example use cases such as a user editing a Word document that is synchronized between the user’s home and work computers. (Sigurdsson, ¶¶26-28 (also describing Server 106 as optional “drop box” “conduit” between client devices); Balakrishnan, ¶470.)

A POSA would have found it obvious that in this type of use case, after the copy of the Word document (which may be called “Version 1”) is transmitted from the user’s home computer to the user’s work computer via Server 106, the user may continue to edit that document on the work computer while the user is at work. (Balakrishnan, ¶471.) Based on Sigurdsson’s bidirectional file synchronization architecture as illustrated in Figure 4, a copy of the updated, edited file at the work

computer (call it “Version 2”) would then be transmitted from the work computer, through Server 106, and back to the user’s home computer. (*Id.*) However, Sigurdsson does not provide details about what would happen at the user’s home computer or Server 106 in that circumstance where “Version 2” of the file is received at Server 106 and the home computer that previously contained “Version 1” of the same file. (*Id.*) A POSA would have been motivated to look for solutions to replace old versions of files with new ones as described in this limitation. (*Id.*) A POSA would have found it obvious that one possibility would be for the updated modified file (“Version 2”) to replace the outdated older version of the file (“Version 1”) on Server 106 and the home computer. (*Id.*) A POSA would have found it obvious to replace the older version of the file for reasons including the obvious benefit of preventing copies of outdated versions from endlessly accumulating in storage on the user’s computers and Server 106. (*Id.*)

Sigurdsson further discloses that the “Synchronization Manager 332” in Server 106 “contains a Space Quota 342” that may be used to “purge” certain client content “whenever the storage space quota for the client is reached.” (Sigurdsson, ¶48; Balakrishnan, ¶472.) Sigurdsson further discloses that content “may no longer exist on the server if it was deleted, for example, due to *retention rules*, storage quotas, or an unforeseen loss of data.” (Sigurdsson, ¶96; Balakrishnan, ¶472.) Perhaps the most obvious example of such a retention rule would be to replace older

versions of files with newer versions, in view of Sigurdsson's dual concerns of maintaining file currency across multiple client devices "when content is changed" (i.e., as soon as possible) (Sigurdsson, ¶84) and not wasting disk space on the server (*id.*, ¶96). (Balakrishnan, ¶472.)

As another example, it would have been obvious to use the storage quota functionality of Server 106 to delete an older version of a file and replace it with an updated version. (Sigurdsson, ¶98; Balakrishnan, ¶473.) For example, Sigurdsson discloses that the Server 106 may have a configurable storage quota for each client it serves. (Sigurdsson, ¶98; Balakrishnan, ¶473.) "If the client's quota is exceeded, the new content can be accepted, but the oldest content can be deleted." (Sigurdsson, ¶98; Balakrishnan, ¶473.) In this example, one obvious use case could be for Server 106 to receive and synchronize multiple versions of a single file (modified first electronic file copy) from Client A until Client A's storage quota is reached, requiring eventual deletion of an old version of the file to make room for the new version. (Balakrishnan, ¶473.) In this use case, the modified first electronic file copy is stored on the memory device associated with the first electronic device in lieu of an older version of the first electronic file copy that was stored on the memory device associated with the first electronic device. (Balakrishnan, ¶473.)

Furthermore, **Shappell** expressly teaches and motivates this feature with its "automatic replication" option. (Balakrishnan, ¶474.) Shappell's "automatic

replication” option further renders obvious implementing Sigurdsson’s Server 106 wherein the modified first electronic file copy is stored in lieu of an older copy of the file. (Balakrishnan, ¶475.) Shappell teaches a user-selectable “File Replication” setting by which 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, ¶475.) When “automatic replication” is enabled, Shappell teaches automatically replacing an “out-of-date file” with an “updated version” on a receiving computer. (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, ¶475.)

In the obvious scenario in which all synchronizing clients implement Shappell’s “automatic replication” feature for a particular file, it would have been obvious that Server 106 should delete all old versions of the file stored on Server 106 after receiving the latest version, given that no user device would need an older version (as older versions get deleted on the client devices), especially in view of

Sigurdsson's storage quota concerns for Server 106. (Balakrishnan, ¶476.) A POSA would have been motivated to incorporate Shappell's auto-replication into each of Sigurdsson's clients to address Sigurdsson's concerns to not waste memory storing old versions of files and 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, ¶476.) The same motivations to combine Sigurdsson and Shappell discussed for claim 1 would apply for this claim. (Balakrishnan, ¶¶182-184, 189, 461-462.)

4. Claim 10.

Claim 10 is substantially identical to claim 1 except that it recites a "**method**" instead of a "**system**." (Balakrishnan, ¶481.) Because claim 10 presents no material difference from claim 1, claim 10 would have been obvious for the same reasons discussed above for claim 1. (Balakrishnan, ¶482.)

5. Claim 12.

This claim is substantively identical to claim 3 and, for the reasons discussed above in connection with claim 3, would have been obvious. (Balakrishnan, ¶483.)

6. Claim 13.

This claim is substantively identical to claim 4 and, for the reasons discussed above in connection with claim 4, would have been obvious. (Balakrishnan, ¶484.)

C. Ground 2: Obviousness of Challenged Claims over Brown in combination with Hesselink.

1. Claim 1.

(a) [1pre]

Brown discloses a client/server “system” shown in Figure 1, including at least two client devices, such as desktop computer **130** and portable computer **135**.

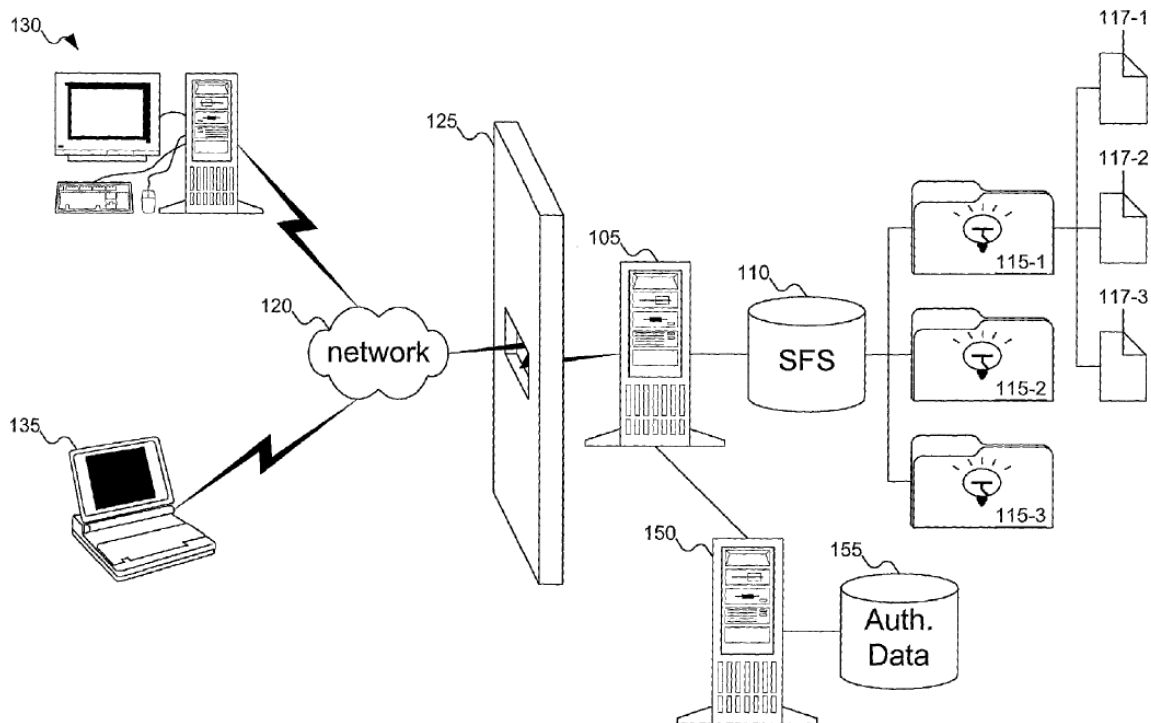


FIG. 1

(Brown, Fig. 1, 2:57-65, 4:20-24, 4:62; Balakrishnan, ¶¶577-580, 803.)

(b) [1a]

In the system of Figure 1, server 105 is a “**first electronic device**,” which includes a “**Synchronization File System (SFS)**” (“**first application**”) that includes a “functional interface and metadata organization that enables efficient and reliable

synchronization by the clients.” (Brown, 2:59-62, 3:56-57; Balakrishnan, ¶¶804-806.) “Typically, server SFS 110 is installed on a hard disk drive within server 105” (Brown, 3:59-60; Balakrishnan, ¶805.)

Server 105 is “**associated with a user**,” at least because the SFS stores folders that are each associated with a user and may only be accessed by a user once the user is logged in. (Brown, 3:62-4:2 (“Stored on server SFS 110 are folders 115–1, 115–2, and 115–3 *Each of the folders is assigned to a user.* Once the user has logged in, the user can access the files and directories (cumulatively called directory entries) within the folder.”); Balakrishnan, ¶806.) Brown further explains that each user has a server “account.” (See Brown, 4:62-64 (one user account may be associated with multiple client devices), 2:65-67, 5:3-4, Balakrishnan, ¶806.)

(c) [1b]

Brown discloses and renders obvious that Brown’s multiple client devices may be in communication with server 105. (Balakrishnan, ¶¶807-823.) For example, in the system depicted in Figure 1, client 130 is an exemplary “**second electronic device**,” and client 135 is an exemplary “**third electronic device**.” (Balakrishnan, ¶¶808, 831.) The server’s SFS communicates with client 130’s Synchronization Applications (“SA”) (“**second application**”) and client 135’s SA (“**third application**”) to facilitate file transfers between the server and clients. (Brown, 3:1-8; Balakrishnan, ¶815.)

Brown teaches that clients 130 and 135 are “**each associated with a user.**” (Brown, 4:62-64 (“multiple clients of a single user account” may “push changes to the server concurrently”), 11:55-56 (“each user can have several clients accessing a single folder”); Balakrishnan, ¶808.) Brown addresses the issue that an individual user may have multiple computers, such as work, home, and portable computers, and may need to “access the same files from the various computers.” (Brown, 1:24-50 (addressing the need to “maintain distributed files across multiple clients, maintaining currency at each client as changes are made”); Balakrishnan, ¶808.) Brown explains that client 130 may be a user’s desktop computer and client 135 may be the same user’s portable computer. (Brown, 4:20-24; Balakrishnan, ¶808.)

Brown in combination with Hesselink discloses and renders obvious that server 105 is configured to “**receive, via a first application at the first electronic device,**” (e.g., the SFS, which handles all synchronization functionality at server 105) “**a copy of a modified first electronic file from a second application at a second electronic device associated with the user**” (client 130’s SA), “**wherein the modified first electronic file copy is automatically received from the second application responsive to the user modifying a content of the first electronic file.**” (Balakrishnan, ¶¶816-823.) For example, Brown describes in detail an exchange of metadata information between client 130 and server 105 to determine file changes on the client and server. (Brown, Figs. 4A-4C, 5, 8A-11F; *id.*, 10:7-9,

10:7-11:52, 11:64-66, 13:57-16:39; Balakrishnan, ¶¶809-811.) Brown also teaches that client 130's SA may monitor activity in its local file system to identify modified files to send to the server. (Brown, 3:47-50 ("The client SA monitors local file system activity within the directory (sometimes called a folder) on the client so that it can efficiently *locate changes to send to the server SFS during synchronization.*"); *see also id.*, 8:13-17 (SA includes a "filter driver 350" that "watches for other applications accessing the files in folder 302, so as to *determine which files on the client have been changed.*"); Balakrishnan, ¶818.) Brown references a "text document" as an exemplary "**first electronic file.**" (Brown, 11:64-66; Balakrishnan, ¶879.)

Upon identifying a modified file, client 130 may send a copy of it to the server when more than a threshold portion of the file has been modified. (Balakrishnan, ¶813.) The threshold inquiry is performed by separating the file into blocks and performing a hash function on each block to determine message digest array (MDA) values. (Brown, 12:15-16, 12:41-45; Balakrishnan, ¶813.) Then, "if less than a threshold number of message digests in the MDAs match, the entire file is transmitted" to the server. (Brown, 12:49-51; *see also id.*, Fig. 11B (steps 1117, 1122, 1120), 15:41-54 (describing that "if a previous version of the file is on the server," and if "an insufficient number of message digests match between the

Petition for *Inter Partes* Review of
U.S. Patent No. 10,067,942

received and computed MDAs,” then at step 1120 the client “*uploads the entire file*”), 2:59-62, 11:22-30; Balakrishnan, ¶¶812-813.)

Brown discloses that such synchronization of modified files may be performed “**automatically**” using the client’s SA to transfer modified files to the server. (Brown, 3:40-47 (“From the user’s perspective, the synchronization process is *automatic*, runs in the background and requires minimal monitoring or intervention.”); Balakrishnan, ¶814.)

In addition, it would have been obvious in view of Brown in combination with Hesselink to automatically receive a copy of a modified file from client 130 at server 105 responsive to the user modifying a content of the first electronic file. (Balakrishnan, ¶¶817-823.) As noted above, Brown teaches that the client SA “can use any scheduling algorithm to initiate synchronization” and that the user can also initiate synchronization “at any time.” (Brown, 3:40-47; Balakrishnan, ¶819.) Brown also seeks to provide “a way to maintain distributed files across multiple clients, maintaining currency at each client *as changes are made*[.]” (Brown, 1:46-50; Balakrishnan, ¶819.) A POSA would have found it obvious in view of these disclosures to look to Hesselink for a technique to serve Brown’s stated purpose of distributing files automatically across multiple clients to maintain file currency at each client and thereby “**automatically**” transfer the “**modified first electronic file**

copy” “from a second application at a second electronic” “responsive to the user modifying a content of the first electronic file.” (Balakrishnan, ¶819.)

Like Brown, Hesselink discloses a synchronization system in which a user may have multiple computers that synchronize versions of a particular file, such as home, work, and laptop computers. (Hesselink, ¶210 (“The present system may also be optionally employed to perform synchronization of files among a user’s multiple computers.”); *id.*, ¶192 (“Thus, the user is ensured of having a completely up to date master copy of each shared data file For example, the system may be set up such that when a user is away, working on his/her laptop, a personal file is automatically updated at the user’s home computer storage device, while a business file is automatically updated at the user’s work computer storage device.”); Balakrishnan, ¶820.)

Hesselink teaches that its system “**automatically**” transmits file changes to other synchronizing devices when the user modifies files. (Balakrishnan, ¶821.) Synchronization occurs “***as soon as the update has been performed***” at one device (for example, a home or office location), so that all connected computers “may be automatically updated concurrently with the updates of the device module file, each time an update is performed.” (Hesselink, ¶183 (“***[W]hen data is updated***, such as when a file is edited, for example, ***the system may be configured to automatically***

update all or some designated subset of *computers on the network* as soon as the update has been performed at one location.”); *id.*, ¶189; Balakrishnan, ¶823.)

Motivation to Combine Brown with Hesselink

It would have been obvious to combine the teachings of Brown and Hesselink, predictably resulting in the synchronization system of Brown 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, 823.)

As an initial matter, Brown and Hesselink are analogous references that are both in the same field as the Challenged Patents. (Balakrishnan, ¶602.) Both references are directed to techniques for effectively synchronizing modified versions of files between client devices over a network. (*Id.*) Both references are reasonably pertinent to the problems faced by the inventor of the challenged patent for improving upon existing file synchronization systems. (*Id.*)

The combination would have been a straightforward matter of combining prior art elements according to known techniques to yield predictable results, namely the system of Brown as modified in view of Hesselink’s teachings as summarized above. (Balakrishnan, ¶603.) The modification would have provided a predictable improvement to Brown’s system, improving the Brown system in the same way that

the relevant features enhance Hesselink's disclosed system, by synchronizing files between clients and a server responsive to the user modifying files. (*Id.*)

In addition, based on the specific teachings of Brown and Hesselink, a POSA would have been motivated to consider Hesselink in view of Brown specifically for the causation of file synchronization. (Balakrishnan, ¶604.) For example, a POSA would have been motivated to consider Hesselink in view of Brown's stated goal to provide "a way to maintain distributed files across multiple clients, maintaining currency at each client as changes are made." (Brown, 1:46-50; Balakrishnan, ¶604.) In addition, Brown would have encouraged a POSA to consider alternative scheduling algorithms for synchronization, including those discussed in Hesselink, given that it expressly teaches that "any" algorithm may be used. (Brown, 3:42-46; Balakrishnan, ¶604.)

Hesselink also provides express motivations to combine its teachings with Brown's to implement Hesselink's automatic concurrent transmission of changes to files, including (1) saving time for the user, and (2) avoiding problems associated with storing two different versions of a file or an outdated version of a file. (Hesselink, ¶189 ("***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.***");

Balakrishnan, ¶605.) These express motivations naturally harmonize with Brown’s stated goal to provide “a way to maintain distributed files across multiple clients, ***maintaining currency at each client as changes are made.***” (Brown, 1:46-50; Balakrishnan, ¶605.) In addition, given this concern in Brown, a POSA would have found it obvious to look to Hesselink’s teachings to automatically push file updates to connected clients as soon as they occur. (Hesselink, ¶183; Balakrishnan, ¶606.)

A POSA would have had at least a reasonable expectation of success in implementing the modified system of Brown in this manner. (Balakrishnan, ¶607.) Brown already discloses that its clients may use any scheduling algorithm to initiate synchronization, as discussed above. (*Id.*) Initiating synchronization responsive to the user modifying the file would have been well within the grasp of a POSA, as well as implementing the modified system of Brown accordingly, using well-known computer software and networking knowledge and design principles. (*Id.*) The references provide ample detail such that a POSA would have been able to implement the modification without undue experimentation. (*Id.*)

A POSA would not have perceived any obstacle or dissuasion from the fact that Hesselink discloses that, in its system, only the changes to a file are transmitted in synchronization, rather than transmitting the entire modified file, which consumes less bandwidth than transferring the entire file. (Balakrishnan, ¶608.) Brown itself teaches that either only the changes, or the entire file, may be transferred, depending

Petition for *Inter Partes* Review of
U.S. Patent No. 10,067,942

on the extent of the changes to the file (whether below or above a threshold). (Brown, 12:61-13:3, Balakrishnan, ¶608.) A POSA would therefore understand that Brown already contemplates and accounts for a balancing and tradeoffs between transferring only the changes and transferring the entire file. (Balakrishnan, ¶608.) For example, a POSA would understand that transferring only changes generally would consume less bandwidth to transfer, but would provide a more complicated approach, requiring tracking of portions of the file that have changed, and would consume more processing time and resources to deal with the differential analysis and reconstitution, whereas transferring the entire file would be a simpler approach. (*Id.*) Thus, a POSA would have understood that Brown already accounts for the possibilities of transferring only changes or transferring an entire file, but Brown goes further than Hesselink with more nuanced details about the relevant functionality that determines whether to transfer the entire file. (*Id.*)

Furthermore, it is immaterial that Hesselink's system transfers only changes given that Hesselink's relevant teaching for purposes of the obviousness inquiry is to synchronize files as changes are made to the files, serving Brown's stated goal of "maintaining currency at each client as changes are made," Brown, 1:46-50, and consistent with Brown's teaching that any synchronization algorithm can be used to initiate synchronization, *id.*, 3:44-46. (Balakrishnan, ¶609.)

In addition, in connection with considering the use of bandwidth to synchronize files, a POSA would have appreciated Hesselink's teaching that its approach of automatically synchronizing changes to a file "among a user's multiple computers" "as the user makes changes to the file" "may be set to apply to only one file or any number of files" at the user's option. (Hesselink, ¶210; Balakrishnan, ¶610.) This teaching naturally harmonizes with Brown's disclosure that files within a certain designated "directory" (or "folder") on each user client device are synchronized, as well as Brown's teaching that any synchronization algorithm may be used, and that synchronization could be initiated manually at any time. (Brown, 6:61-7:20, 3:44-50; Balakrishnan, ¶610.) A POSA would have appreciated Hesselink's teaching and motivation that Brown's synchronization system could apply different synchronization algorithms to different subsets of files in the monitored synchronized directory to differentiate between different needs and values associated with different files or types of files and account for different bandwidth availabilities. (Brown, 12:4; Hesselink, ¶¶3-4, 12 (noting the availability of high-speed and reliable network connections such as Ethernet, where a "typical Ethernet local area network can facilitate 100 mbps for both upload and download data transfer rate" and noting that "high bandwidth" data networks "can be provided at relatively low cost"); Balakrishnan, ¶610.) For example, a user with fast and reliable Ethernet network connections for their client devices could designate either

their entire synchronized directory, or a subset of a particular time-sensitive or important work file(s) (or subdirectory of files), to be automatically synchronized as changes are made to the file(s) as described in Hesselink, while optionally leaving the remainder of less time-sensitive or less important files in the user's synchronized directory to be synchronized only on a periodic basis. (Balakrishnan, ¶610.) A POSA applying Hesselink's teachings to Brown would be motivated to design the system to permit different synchronization algorithms for different subsets of files in this regard. (Balakrishnan, ¶610.)

In addition, a POSA would not have been dissuaded from combining Hesselink with Brown in view of Hesselink's use of a peer-to-peer model as opposed to Brown's client/server model. (Balakrishnan, ¶611.) Hesselink aimed to improve upon certain features of prior art systems. (*Id.*) For example, Hesselink discusses then-current solutions that required "time consuming," "tedious," and manual synchronization processes that involved uploading files to a "central server" and were "cumbersome and time consuming to use," as well as "virtual drives" that had their own drawbacks. (Hesselink, ¶¶8-9, 22; Balakrishnan, ¶611.) A POSA would have recognized, however, that Brown's system provides a user-friendly, automatic synchronization approach that does not suffer from the perceived drawbacks of these prior art solutions. (Brown, 3:40-42 ("From the user's perspective, the synchronization process is automatic, runs in the background and requires minimal

Petition for *Inter Partes* Review of
U.S. Patent No. 10,067,942

monitoring or intervention.”); Balakrishnan, ¶¶611.) Hesselink does not specifically disparage or motivate away from applying its relevant synchronization scheduling function to a client/server model. (Balakrishnan, ¶¶611.) Notably, Hesselink’s summary of the “need” in the prior art and its “Summary of the Invention” section do not specifically discuss or distinguish the form of network model as between peer-to-peer or client/server. (Hesselink, ¶¶23-34; Balakrishnan, ¶¶611.) A POSA would have considered Hesselink’s relevant teachings to be applicable to improve upon Brown’s user-friendly, automatic client/server synchronization systems as well, in view of Hesselink’s express motivation to serve the stated goals of Brown of maintaining currency among a user’s multiple devices, and Brown’s suggestion that any synchronization scheduling algorithm may be used. (Balakrishnan, ¶¶611.) Hesselink also does not teach that the devices in its synchronization network must not include any servers. (*Id.*) To the contrary, Hesselink specifically teaches that its disclosed system may include servers among the computers that are synchronized: “As noted above, any of computers 72, 74 and 76 may be a desktop computer, laptop computer, personal digital assistant, network attached storage system, *file server* or other computing device that has CPU, storage and connectivity capabilities.” (Hesselink, ¶122; Balakrishnan, ¶¶611.) A POSA would have understood that a “file server” in this context would be a server for storing files and making the files available to client devices, much like the server in the Brown

Petition for *Inter Partes* Review of
U.S. Patent No. 10,067,942
client/server system. (Balakrishnan, ¶611.)

(d) [1c]

This limitation would have been obvious in view of Brown in combination with Hesselink. (Balakrishnan, ¶¶824-827.) For example, it would have been obvious to synchronize the server with a client device, for example laptop computer 135 (“**third electronic device**”) described in Brown. (Balakrishnan, ¶825.) The client would attempt to initiate communication with the server, and the server could “**determine whether the first electronic device**” (server 105) “**is in communication with a third electronic device**” (client 135) based on whether the server receives the client’s initial communication. (Brown, 3:1-8, 3:42-46; Balakrishnan, ¶825.)

In addition, this limitation would have been obvious in view of Hesselink. (Balakrishnan, ¶¶826-827.) Hesselink discloses repeatedly checking whether devices are in communication as part of synchronization. (Hesselink, ¶¶182, 190 (“[T]he system is persistent and may be configured on a user computer to automatically and repeatedly attempt to reconnect with any and all remote devices which are actively subscribed to when a connection is broken during an active subscription.”), 196; Balakrishnan, ¶826.) It would have been obvious to use this functionality from Hesselink in Brown’s system to address situations in which client 135 is not currently online, which addresses Brown’s goal of maintaining file-update

currency among a user's synchronizing devices. (Balakrishnan, ¶¶826-827.) In the Brown system as obviously modified by Hesselink, when client 135 is not currently online, it would have been obvious to periodically check whether communication with client 135 has resumed, to determine whether files may be transferred to the client. (Hesselink, ¶183 (“[W]hen data is updated, such as when a file is edited, for example, the system may be configured to automatically update all or some designated subset of computers on the network as soon as the update has been performed at one location. *Computers that are offline at such a time will be automatically updated as soon as those computers go online again.* The persistence of the system ensures that updates are completed for all online computers, as if there is some failure or problem during the update, the connection restores itself to ensure that the update is completed.”); Balakrishnan, ¶¶827.) The same motivations to combine Brown and Hesselink discussed for claim [1b] would apply for this limitation. (Balakrishnan, ¶¶601-611.)

(e) [1d]

This claim limitation would have been obvious in view of Brown in combination with Hesselink. (Balakrishnan, ¶¶828-835.) For example, it would have been obvious to “**configure**” Brown's SFS associated with server 105 (“**first application**”) to “**automatically send, via the first application, the modified first electronic file copy to a third application**” (e.g., client 135's SA) “**at the third**

electronic device” (client 135). (Balakrishnan, ¶¶829-835.) As discussed above for claim [1b], each of Brown’s clients includes an SA that is an application that sends and receives files to and from the SFS. (Balakrishnan, ¶¶807-823.) After execution of the functionality discussed for claim [1b], above, client 135 would initiate communication with the server and receive a copy of the modified first electronic file sent to server 105 by client 130, as described above for claim [1b]. (Brown, 2:63-67, 3:1-8; Balakrishnan, ¶¶829-830.)

A POSA would have found it obvious to “**automatically**” send the modified first electronic file copy to laptop client 135 “**responsive to receiving the modified first electronic file copy from the second electronic device,**” particularly in view of Hesselink’s teachings discussed above for claim [1b] regarding automatically and immediately forwarding file updates to all synchronizing clients. (Balakrishnan, ¶¶807-823, 831.) As discussed above for claim [1b] and [1c], Brown’s clients include similar SAs that operate according to the same synchronization algorithm; therefore, Brown’s clients operate similarly and are interchangeable. (Balakrishnan, ¶¶807-827, 830, 837.) Thus, it would have been obvious that Brown’s synchronization process on laptop client 135 would begin by checking whether the server has any updates, including updates from other client devices, such as desktop client 130. (Brown, 10:18-27; Balakrishnan, ¶831.) Thereafter, in the same, automated manner as client 130, client 135 could “download[] all the pertinent

Petition for *Inter Partes* Review of
U.S. Patent No. 10,067,942

changes from server 105.” (Brown, 10:53-55; Balakrishnan, ¶831.) In addition, for the reasons above for claim [1c], it would have been obvious to only transfer updated files to client 135 “**responsive to the determination that the first electronic device**” (server 105) “**is in communication with the third electronic device**” (client 135). (Balakrishnan, ¶¶824-827.) The same motivations to combine Brown and Hesselink discussed for claim [1b] would apply for this limitation. (Balakrishnan, ¶¶601-611.)

(f) [1e]

This limitation would have been obvious in view of Brown in combination with Hesselink. (Balakrishnan, ¶¶836-838.) It would have been obvious to implement the functionality performed by client 130 in claim [1b] but instead using the SA (“**third application**”) at client 135 (“**third electronic device**”) to automatically transfer a “**copy of a modified second electronic file**” to the server “**responsive to the user modifying a content of the second electronic file.**” (Balakrishnan, ¶¶807-823, 837.) As discussed above for claim [1b], client 135 is obviously “**associated with the user.**” (Balakrishnan, ¶¶807-823.) A POSA would have found it obvious that a “text document” that is different from the “**first electronic file**” could be an exemplary “**second electronic file.**” (Brown, 11:64-66; Balakrishnan, ¶879.)

Brown teaches that synchronization is bidirectional between multiple client devices that “synchronize with each other by synchronizing to a common server account” and use the same “Synchronization Application (SA).” (Brown, 2:63-67; Balakrishnan, ¶837.) A POSA thus would have understood from Brown’s disclosures that file changes made on either of client devices 130 or 135 would have been synchronized to other synchronizing client devices in the same way through server 105, such that clients 130 and 135 operate similarly and interchangeably. (Balakrishnan, ¶837.)

A POSA would have understood that Brown’s client devices 130 and 135 are fungible in the sense that files would be synchronized bidirectionally between the client devices as files are modified on both devices. (Balakrishnan, ¶837.) For example, a POSA would have understood and found it obvious that a “**second electronic file**” on the “**third electronic device**” (client 135) would have been transferred via client 135’s SA application (“**third application**”) to server 105 when that file is modified. (Balakrishnan, ¶838.) Brown describes a synchronization process with respect to desktop client 130 that starts by checking whether the server has any updates for the user’s account, which would include updates from other devices of the user such as laptop client 135. (Brown, 10:18-27; Balakrishnan, ¶838.) Later, client 130 can send all pertinent changes made on client 130 to server 105 to update other client devices, including client 135, as would be motivated by a

user wanting to keep, for example, work and home computers synchronized as discussed in Brown. (Brown, 10:53-55; Balakrishnan, ¶838.) Because the synchronization process proceeds similarly from the perspective of clients 130 and 135, this limitation would have been obvious in view of Brown in combination with Hesselink for the reasons discussed above for claim [1b]. (Balakrishnan, ¶838.) The same motivations to combine Brown and Hesselink discussed for claim [1b] would apply for this limitation. (Balakrishnan, ¶¶601-611.)

(g) [1f]

For the reasons discussed above for claim [1d] and [1e], Brown's clients operate similarly and are interchangeable, so it would have been obvious to implement the functionality described in claim [1c] for determining whether the first electronic device is in communication with the third electronic device but instead for **“determin[ing] whether the first electronic device is in communication with the second electronic device.”** (Balakrishnan, ¶¶824-838, 839-840.) Each of Brown's clients undergoes the same synchronization process, including determinations related to communication between a client and server, as discussed above. (Balakrishnan, ¶840.) Thus, for the same reasons above for claim [1c], Brown discloses that its server may detect whether any of its clients are online or offline and in communication or out of communication with the server. (Balakrishnan, ¶824-827, 840.) In addition, this element would have been obvious

Petition for *Inter Partes* Review of

U.S. Patent No. 10,067,942

in view of Brown in combination with Hesselink for the reasons above for claim [1c]. (Balakrishnan, ¶¶824-827.) The same motivations to combine Brown and Hesselink discussed for claim [1b] would apply for this limitation. (Balakrishnan, ¶¶601-611.)

(h) [1g]

This claim limitation would have been obvious in view of Brown in combination with Hesselink. (Balakrishnan, ¶841.) For the reasons discussed above, Brown’s clients operate similarly and are interchangeable. (Balakrishnan, ¶842.) Thus, for the reasons discussed above in connection with claim [1d], it would have been obvious to automatically transfer the modified “**second electronic file copy**” from Brown’s server, via the server’s “**first application**,” to the “**second application**” “**responsive to the determination that the first electronic device**” (server) “**is in communication with the second electronic device**” (desktop client 130) “**and responsive to receiving the modified second electronic file copy from the third electronic device**” (laptop client 135). (Balakrishnan, ¶842.) The same motivations to combine Brown and Hesselink discussed for claim [1b] would apply for this limitation. (Balakrishnan, ¶¶601-611.)

(i) [1h]

Brown teaches that client 135 could “**replace**” “**an older version of the first electronic file**” stored on client 135 “**with the modified first electronic file copy**”

such that the modified file copy is stored in lieu of the older version. (Brown, 13:18-23 (“[I]f less than a threshold number of message digests in the MDAs match, then *the entire file is downloaded from the server*. Once the download file is completely constructed, the client inserts the download file into its final location, *replacing an older version of the file if it exists*.”), 2:63-67, 10:18-27, 10:53-55; Balakrishnan, ¶¶843-845.)

(j) [1i]

For the reasons discussed above, Brown’s clients operate similarly and are interchangeable. (Balakrishnan, ¶¶846-847.) Thus, for the reasons discussed above for claim [1h], it would have been obvious to implement the functionality of claim [1h] but with respect to a “**modified second electronic file copy**” instead of a modified first electronic file copy and a “**second electronic device**” (client 130) instead of a third electronic device (client 135). (Balakrishnan, ¶847.)

2. Claim 3.

Brown discloses that server 105’s SFS (“**first application**”) is “**configured to**” “**store**” “**the modified first electronic file copy to a memory device associated with the first electronic device.**” (Balakrishnan, ¶¶848-849.) For example, the modified first electronic file copy may be stored to a folder associated with the user on server 105. (Brown 3:54-4:3 (“[S]erver 105 includes server SFS 110. Server 105 includes all the typical elements of a server, such as a central processor,

memory, bus, disk space, etc. Typically, server SFS 110 is installed on a *hard disk drive* within server 105, but a person skilled in the art will recognize other forms of media usable by server 105: for example, removable media or optical media. *Stored on server SFS 110 are folders 115–1, 115–2, and 115–3. . . . Each of the folders is assigned to a user.* Once the user has logged in, the user can access the files and directories (cumulatively called directory entries) within the folder.”), 4:64-5:1, 13:1-3; Balakrishnan, ¶¶849.)

3. Claim 4.

As discussed above in connection with claim 3, Brown discloses that “**the modified first electronic file copy is stored on the memory device associated with the first electronic device.**” (Balakrishnan, ¶¶850-862.) Brown discloses and renders obvious that the server SFS will store the modified file in memory in lieu of an older version of the file that is “replaced” by the modified file: “Access to the server SFS is restricted to file level operations via the protocol where new files can be uploaded, *existing files can be* downloaded, *replaced*, renamed, moved, or deleted, but an existing file cannot be modified directly.” (Brown, 4:44-47; Balakrishnan, ¶852.) A POSA would have found it obvious in view of this disclosure to replace old versions of files with new ones at the server when instructed to do so via the aforementioned file level operation. (Balakrishnan, ¶852.)

Motivation to do so would have included the conservation of memory storage resources via the avoidance of storing an endless set of old versions of files. (*Id.*)

In addition, a POSA would have found it obvious that where a client uploads a modified version of a file to the server, the teaching that “existing files can be . . . replaced” on the server would obviously mean that an older version of the file on the server would be replaced by the uploaded modified file, which would be stored on the server in place of the older version. (Balakrishnan, ¶853.) Brown describes that when the client uploads a modified file to the server, the server checks for “the version of the file on the server” and then uploads the file (which will upload the complete file, unless a sufficient number of MDA values match). (Brown, 12:44-13:3; Balakrishnan, ¶853.) The server then “constructs the new version of the file” and “updates the file metadata.” (Brown, 12:65-13:3; Balakrishnan, ¶853.) Brown explains:

Before the client SA uploads a file, it computes an MDA from the file. It then requests from the server the MDA for ***the version of the file on the server*** by sending to the server the SID of the file, the name of the file, and the directory to which the file is to be uploaded. The server then checks to see if it has a file with that SID or if there is a file with the same name as that specified by the client in the directory to which the client is uploading the file. If the server finds a version of the file, it returns the file’s MDA to the client. The client SA compares the two MDAs and if a sufficiently high number of message digests match, it performs a special upload where only the differing message digests and their corresponding 4K data blocks are uploaded. ***The server constructs the new version of the file*** by starting with a copy of the previous version and modifying it with the uploaded data. Once the file has been

completely uploaded, *the server then stores the file in the specified directory and updates the file metadata.*

(*Id.*, 12:53-13:3.) A POSA would have found it obvious that this operation is an example of where “existing files can be . . . replaced” on the server. (Balakrishnan, ¶853.)

Furthermore, Brown’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 computer, work computer, and the server and the issue of synchronizing all three devices to make sure they are updated with the most recent updates on each respective version. (Balakrishnan, ¶854.) For example, Brown expressly references a client’s desktop computer and laptop computer as two exemplary “clients,” and describes bidirectional synchronization between such devices via a common server account: “Client 130 represents an ordinary desktop computer, as might be present at a user’s place of work or home. Portable computer 135 represents a notebook or laptop computer, as a user might take with him to a hotel room on business.” (Brown, 4:20-24, 1:62-63 (“Client machines synchronize with each other by synchronizing to a common server account.”); Balakrishnan, ¶854.)

A POSA would have found it obvious that in this type of use case, after the copy of a file stored on the user’s desktop computer (which may be called “Version 1”) is transmitted from the user’s desktop computer to the user’s laptop computer via the server, the user may continue to edit that document on the laptop computer

while the user is on a business trip. (Balakrishnan, ¶855.) Based on Brown’s bidirectional file synchronization architecture, a copy of the updated, edited file at the laptop computer (call it “Version 2”) would then be transmitted from the laptop computer, through the server, and back to the user’s desktop computer. (*Id.*) A POSA would have been motivated to look for solutions to replace old versions of files with new ones in this use case to conserve memory storage resources. (*Id.*) A POSA would have found it obvious that one possibility would be for the updated modified file (“Version 2”) to replace the outdated older version of the file (“Version 1”) on the server and the desktop computer. (*Id.*) A POSA would have found it obvious to replace the older version of the file for reasons including the obvious benefit of preventing copies of outdated versions from endlessly accumulating in storage on the user’s computers and the server. (*Id.*)

4. Claim 10.

Claim 10 is substantially identical to claim 1 except that it recites a “**method**” instead of a “**system**.” (Balakrishnan, ¶¶863-864.) Because claim 10 presents no material difference from claim 1, claim 10 would have been obvious for the same reasons discussed above for claim 1. (Balakrishnan, ¶864.)

5. Claim 12.

This claim is substantively identical to claim 3 and, for the reasons discussed above in connection with claim 3, would have been obvious. (Balakrishnan, ¶865.)

6. Claim 13.

This claim is substantively identical to claim 4 and, for the reasons discussed above in connection with claim 4, would have been obvious. (Balakrishnan, ¶866.)

IX. CONCLUSION

Petitioners respectfully request IPR institution.

Petition for *Inter Partes* Review of
U.S. Patent No. 10,067,942

Dated: January 4, 2023

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

Respectfully submitted,

By: /Heidi L. Keefe/
Heidi L. 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

Petition for *Inter Partes* Review of
U.S. Patent No. 10,067,942

APPENDIX A

Petition for *Inter Partes* Review of
U.S. Patent No. 10,067,942

Claims Listing
Claims 1, 3, 4, 10, 12 and 13
U.S. Pat. No. 10,067,942

Claim 1:

[1pre] A system comprising:

[1a] a first electronic device, associated with a user, configured to:

[1b] receive, via a first application at the first electronic device, a copy of a modified first electronic file from a second application at a second electronic device associated with the user, wherein the modified first electronic file copy is automatically received from the second application responsive to the user modifying a content of the first electronic file;

[1c] determine whether the first electronic device is in communication with a third electronic device;

[1d] automatically send, via the first application, the modified first electronic file copy to a third application at the third electronic device responsive to the determination that the first electronic device is in communication with the third electronic device and responsive to receiving the modified first electronic file copy from the second electronic device;

[1e] receive, via the first application, a copy of a modified second electronic file from the third application at the third electronic device associated with the user, wherein the modified second electronic file copy is automatically received

Petition for *Inter Partes* Review of
U.S. Patent No. 10,067,942

from the third application responsive to the user modifying a content of the second electronic file;

[1f] determine whether the first electronic device is in communication with the second electronic device; and

[1g] automatically send, via the first application, the modified second electronic file copy to the second application at the second electronic device responsive to the determination that the first electronic device is in communication with the second electronic device and responsive to receiving the modified second electronic file copy from the third electronic device,

[1h] wherein, responsive to sending the modified first electronic file copy to the third electronic device, an older version of the first electronic file stored on the third electronic device is automatically caused to be replaced with the modified first electronic file copy such that the modified first electronic file copy is stored on the third electronic device in lieu of the older version of the first electronic file, and

[1i] wherein, responsive to sending the modified second electronic file copy to the second electronic device, an older version of the second electronic file stored on the second electronic device is automatically caused to be replaced with the modified second electronic file copy such that the modified second electronic file copy is stored on the second electronic device in lieu of the older version of the second electronic file.

Petition for *Inter Partes* Review of
U.S. Patent No. 10,067,942

Claim 3: The system of claim 1, wherein the first electronic device is configured to: store, via the first application, the modified first electronic file copy to a memory device associated with the first electronic device.

Claim 4: The system of claim 3, wherein the modified first electronic file copy is stored on the memory device associated with the first electronic device in lieu of an older version of the first electronic file copy that was stored on the memory device associated with the first electronic device.

Claim 10:

[10pre] A method comprising:

[10a] receiving, via a first application at a first electronic device, a copy of a modified first electronic file from a second application at a second electronic device associated with a user, wherein the modified first electronic file copy is automatically received from the second application responsive to the user modifying a content of the first electronic file;

[10b] determining whether the first electronic device is in communication with a third electronic device;

[10c] automatically sending, via the first application, the modified first electronic file copy to a third application at the third electronic device responsive to the determination that the first electronic device is in communication with the third electronic device and responsive to receiving the modified first electronic file copy from the second electronic device;

[10d] receiving, via the first application, a copy of a modified second electronic file from the third application at the third electronic device associated with the user, wherein the modified second electronic file copy is automatically received from the third application responsive to the user modifying a content of the second electronic file;

[10e] determining whether the first electronic device is in communication with the second electronic device; and

[10f] automatically sending, via the first application, the modified second electronic file copy to the second application at the second electronic device responsive to the determination that the first electronic device is in communication with the second electronic device and responsive to receiving the modified second electronic file copy from the third electronic device,

[10g] wherein, responsive to sending the modified first electronic file copy to the third electronic device, an older version of the first electronic file stored on the third electronic device is automatically caused to be replaced with the modified first electronic file copy such that the modified first electronic file copy is stored on the third electronic device in lieu of the older version of the first electronic file, and

[10h] wherein, responsive to sending the modified second electronic file copy to the second electronic device, an older version of the second electronic file stored on the second electronic device is automatically caused to be replaced with the modified second electronic file copy such that the modified second electronic

Petition for *Inter Partes* Review of
U.S. Patent No. 10,067,942

file copy is stored on the second electronic device in lieu of the older version of the second electronic file.

Claim 12: The method of claim 10, further comprising: storing, via the first application, the modified first electronic file copy to a memory device associated with the first electronic device.

Claim 13: The method of claim 12, wherein the modified first electronic file copy is stored on the memory device associated with the first electronic device in lieu of an older version of the first electronic file copy that was stored on the memory device associated with the first electronic device.

Petition for *Inter Partes* Review of
U.S. Patent No. 10,067,942

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 13,343 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 4, 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 L. 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

Petition for *Inter Partes* Review of
U.S. Patent No. 10,067,942

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. 10,067,942**, including all exhibits (**Nos. 1001-1028**) and related documents, are being served via Federal Express on the 4th 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
501 Congress Ave
Suite 150
Austin, TX 94201

DATED: January 4, 2023

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