

Petition for *Inter Partes* Review of
U.S. Patent No. 10,754,823

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. IPR2023-00432

U.S. Patent No. 10,754,823

Filing Date: January 23, 2020

Issue Date: August 25, 2020

Title: Pre-File-Transfer Availability Indication Based on Prioritized Metadata

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 10,754,823**

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 '823 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 over Sigurdsson with Shappell and Rao.....	12

Table of Contents

(continued)

	Page
C. Ground 2: Obviousness over Brown with Hesselink, Shappell, and Rao.....	48
IX. Conclusion	71

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”)

Petition for *Inter Partes* Review of
U.S. Patent No. 10,754,823

List of Exhibits

Exhibit No.	Description of Document
1025	U.S. Patent No. 7,035,847 B2 to David K. Brown, et al. (“Brown”)
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. 10,754,823

I. INTRODUCTION

Petitioners Box, Inc., and Dropbox, Inc., respectfully petition for *Inter Partes* Review (“IPR”) of claims 1-4, 8-11, and 13-16 (“Challenged Claims”) of U.S. Patent No. 10,754,823 (“’823”).

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

Petitioners are also filing IPR petitions challenging the asserted claims of five related patents PO also asserts against Petitioners in litigation: U.S. Patent Nos. 9,143,561; 10,067,942; 10,289,607; 10,642,787; and 11,003,622 (“Related

Petition for *Inter Partes* Review of
U.S. Patent No. 10,754,823

Patents”). All of the patents purport to claim priority to the same provisional and non-provisional applications, and all share substantially similar specifications including substantially the same Detailed Description section.

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

PO’s complaints asserting infringement of the ’823 were served on Box and Dropbox on January 5, 2022. This Petition is timely filed.

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

Petitioners provide the following designation of counsel.

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

Petition for *Inter Partes* Review of
U.S. Patent No. 10,754,823

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

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

III. FEE PAYMENT (§ 42.103)

Petitioners request review of 12 claims. Payment of \$41,500 is submitted herewith.

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

A. Grounds for Standing

Petitioners certify that the '823 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	1-4, 8-11, and 13-16	Obvious by <u>Sigurdsson</u> in view of <u>Shappell</u> and <u>Rao</u>
2	1-4, 8-11, and 13-16	Obvious by <u>Brown</u> in view of <u>Hesselink</u> , <u>Shappell</u> , and <u>Rao</u>

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

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

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

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

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

§325(d): The presented references were not cited during prosecution and are not cumulative of cited references. (Balakrishnan, ¶¶51-52.)

V. OVERVIEW OF '823 PATENT

A. Claim Listing

Appendix A lists the Challenged Claims.

B. Specification

The '823 relates to file transfer and synchronization including "automatic modification-triggered transfer of a file among two or more computer systems associated with a user" between a user's multiple electronic devices. ('823, 1:24-26, 3:55-57, 8:32-39.) Associated metadata may also "cause[] a graphical availability indication of the updated version of the file to be presented (e.g.,

Petition for *Inter Partes* Review of
U.S. Patent No. 10,754,823

proximate a file icon representing the file) at the second client device based on the metadata.” (*Id.*, 4:3-18.) *Cf.* Unified ID at 7-9 (summarizing specification).

VI. LEVEL OF ORDINARY SKILL IN THE ART

As discussed below, the claims’ earliest effective filing date is November 10, 2008. A 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. 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 ’823 claims priority to 11/9/2007 Provisional Application No. 60/986,896 (EX1015) and a 11/10/2008 non-provisional. Patent claims are **not** presumptively entitled to the benefit of a provisional application filing date; PO bears the burden

Petition for *Inter Partes* Review of

U.S. Patent No. 10,754,823

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); *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]-[1e], [8a]-[8d], and [13a]-[13d],² so the claims’ effective priority is no earlier than 11/10/2008. (Balakrishnan, ¶¶65-73.) 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)
Rao	2/22/2005	8/24/2006	§102(a)/(b)/(e)
Shappell	10/24/2003	4/28/2005	§102(a)/(b)/(e)
Brown	3/15/2002	12/19/2002	§102(a)/(b)/(e)
Hesselink	11/13/2004	6/2/2005	§102(a)/(b)/(e)

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** and **Rao**.

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** with **Hesselink**, disclosing the core synchronization features, and **Shappell** and **Rao**.

C. Summary of Prior Art

1. Sigurdsson

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

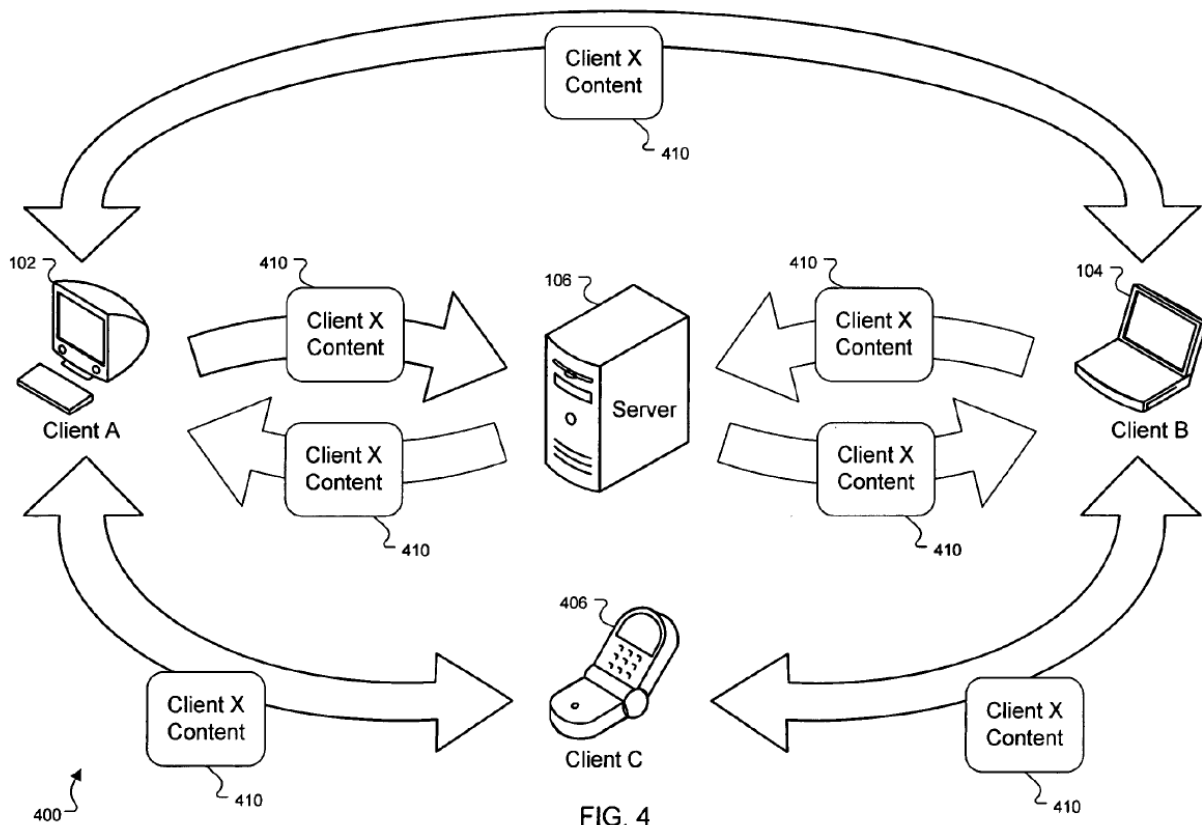


FIG. 4

Clients A, B, and C may be associated with the same user and communicate with a server “through a network,” such as the Internet, to transfer new and modified files to the server. (Sigurdsson, ¶¶58, 69, 84, 112-13, 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).) “Server 106” in Figure 4 “can provide a temporary storage location, or ‘drop box’, for a user’s files, documents, web pages, or the contents of these documents.” (Sigurdsson, ¶27.) Content sent from clients to Server 106 may include copies of files and metadata. (Sigurdsson, ¶54.) A “priority algorithm” may prioritized data items for

synchronization, based on factors such as “size, type, or age.” (Sigurdsson, ¶58.)

Metadata may be sent to Server 106 prior to associated files, such as when the files are large. (Sigurdsson, ¶72.) Server 106 can also send a list of “missing” content, including the metadata, to a client that was offline when the missing content was created or modified. (*Id.*)

2. Rao

Rao teaches synchronizing files between a client and a server in two phases: first synchronizing metadata associated with files and then synchronizing the raw file data. (Rao, ¶63.) Rao explains that transmitting metadata first, before raw data, is useful because it may enable certain functionality before the raw data is present, and the “raw data may be extraordinarily large” and “may require a long time to transfer.” (Rao, ¶¶43-44, 46, 48.) Metadata may be used to display information about a file that is available on the server not on the client. (Rao, ¶43.)

3. Shappell

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

of shared files with new versions when the files are modified. (Shappell, ¶81.)

4. Brown

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

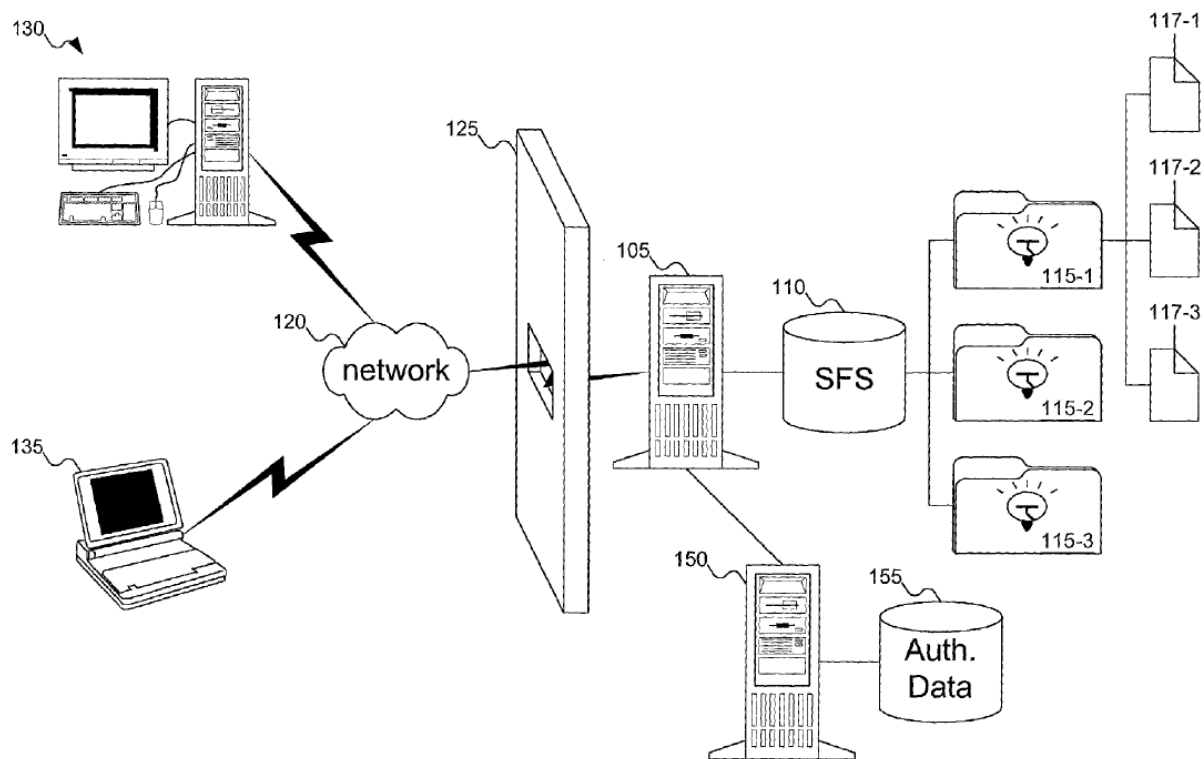


FIG. 1

(Brown, Fig. 1.) “From the user’s perspective, the synchronization process is automatic, runs in the background and requires minimal monitoring or intervention.”

(Brown, 3:40-42.) Brown’s synchronization process includes two main steps: first,

metadata is exchanged between clients and the server, and second, new or recently updated files are exchanged. (Brown, 5:4-24.)

5. Hesselink

Like Brown, Hesselink discloses a synchronization system involving multiple user computers, such as home, work, and laptop computers. (Hesselink, ¶210.) The system allows a user to be “ensured of having a completely up to date master copy of each shared data file” on each user device. (Hesselink, ¶192.) Hesselink teaches automatically synchronizing a file “as soon as the update has been performed” at one device, so all connected devices “may be automatically updated concurrently.” (Hesselink, ¶¶183, 189, 197). Hesselink’s system “automatically and repeatedly” checks whether synchronizing devices are in communication prior to initiating synchronization. (Hesselink, ¶¶182, 189-190.)

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

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.

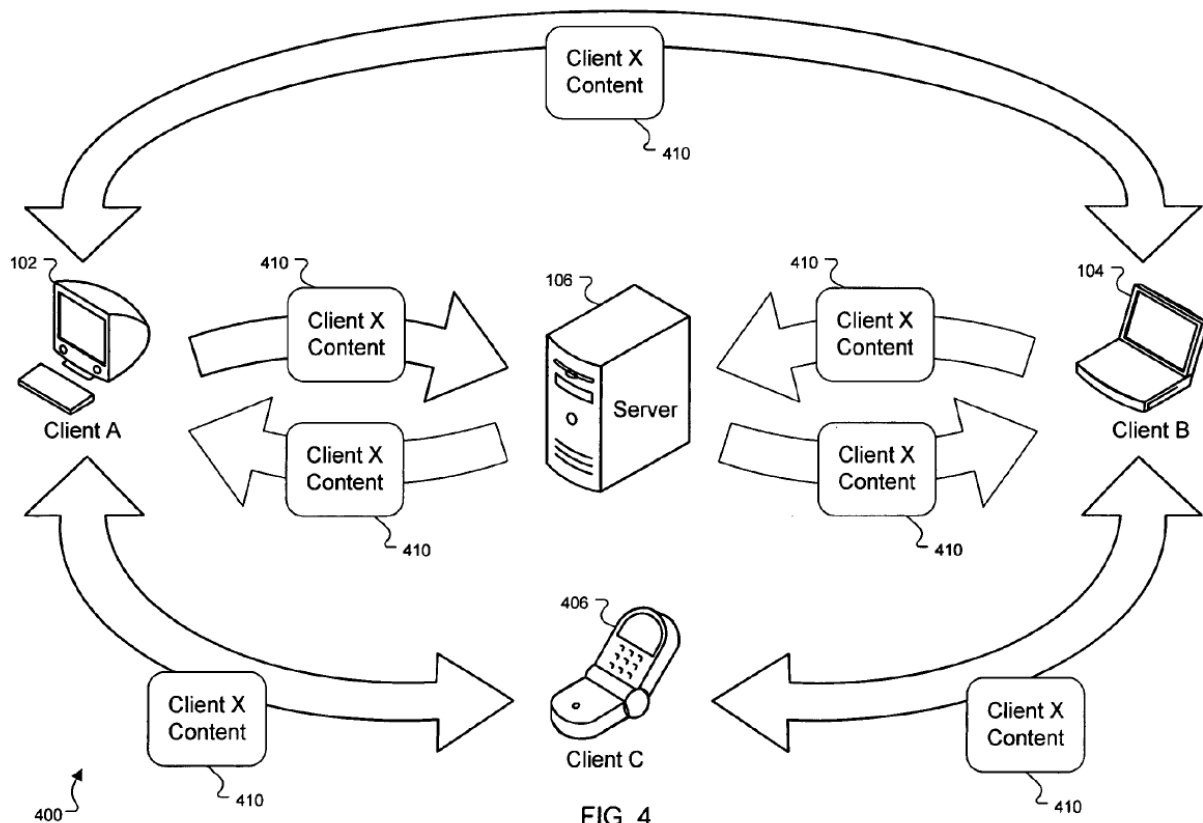


FIG. 4

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

(b) [1a]

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

(c) [1b]

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

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

Server 106 “**automatically receives**” a “**copy of the first file from the first client device responsive to the user modifying a content of the first file.**” (Sigurdsson, ¶¶26-28 (when user saves modifications to file, “Client A Content 108, which can include *a copy of the user’s file*, is *sent automatically* to the Server 106.”), 52 (“edit and save”), 84 (“Client A 102 posts Client A Content 108 to the Server 106 *when content is changed* on the Client A 102” and “posting can occur as soon as an event is generated by an action, such as saving a file”);

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

Although Figure 4 of Sigurdsson depicts both client/server and peer-to-peer connections, a POSA would have appreciated Sigurdsson’s teachings of the obvious advantages of a client/server arrangement. (Balakrishnan, ¶¶154-155.) The server can stay online while user computers periodically go offline, so that Server 106 beneficially provides a “drop-box” for “a user’s files” to **“facilitate sharing this information between the user’s computers by storing the shared content if the target computer is offline.”** (Sigurdsson, ¶¶27-29, 61 (server tracks clients’ online/offline status); Balakrishnan, ¶155.) A server may also have more available bandwidth relative to clients for transferring data. (*id.*; Sigurdsson, ¶79.) 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 (“Although FIG. 4 does not show explicit arrows between

Petition for *Inter Partes* Review of
U.S. Patent No. 10,754,823

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

(d) [1c]

Sigurdsson with Shappell and Rao renders obvious claim [1c]. (Balakrishnan, ¶¶160-194.) Sigurdsson discloses **first metadata associated with the first file generated from the user modifying the content of the first file**. Sigurdsson explains that when the user (e.g., at Client A) modifies a file (e.g., a resume), “[m]etadata describing the content, such as the time the resume was saved, can also be transmitted to the Server 106” along with the file. (Sigurdsson, ¶¶32-34, 5 (“The method also includes automatically extracting content from the first file in response

to the event using the first client and *generating metadata to associate with the content*[.]”), 65 (“metadata that describes . . . when the data was last changed”), 72-73 (“list of the content that was generated” while Client C was offline, such as “metadata describing the file and the time it was accessed”);⁴ Balakrishnan, ¶163.)

Server 106 then **receives over a network, from the first client device** (e.g., Client A), that **first metadata**, along with the updated first file (per claim 1[b]). (Sigurdsson, ¶¶6 (“*transmitting the content and the metadata to the server.*”), 35 (“Server 106 updates database tables with the Client A Content 108 *and any*

⁴ Although this exemplary metadata is disclosed in the context of a “peer-to-peer content flow” (Sigurdsson, ¶65), it would have been obvious to incorporate such metadata into Sigurdsson’s client-server flow, such as the paragraphs 72-73 example where Client A transfers metadata associated with a large file (e.g., maintenance manual) to the server. *Boston Sci. Scimed v. Cordis*, 554 F.3d 982, 991 (Fed. Cir. 2009) (“Combining two embodiments disclosed adjacent to each other in a prior art patent does not require a leap of inventiveness.”); Sigurdsson, ¶¶51, 65, 72; Balakrishnan, ¶163 n.10. For example, a POSA would have been motivated to include metadata about when a large maintenance manual (“**first file**”) was last modified so that users know if they have the most up-to-date maintenance instructions. (Balakrishnan, ¶163 n.10.)

Petition for *Inter Partes* Review of
U.S. Patent No. 10,754,823

corresponding metadata describing the Content 108 it received from the Client

A 102.”); 54 (“In step 506, the content originating on the client is posted to the server. The content can include *a copy of the document* . . . that was saved by the user as well as *metadata that describes information, such as the type of data, its source, when it was saved, and when it was posted to the server*”), 72-73 (list containing generated metadata “*transmitted to the Server*”); Balakrishnan, ¶164.)

Claim [1c] further recites “**the first metadata being assigned a *priority* that is greater than a second *priority* assigned to the copy of the first file,**” and claims [1d], [1h], and [1i] recite further “priority”-related limitations. (Balakrishnan, ¶165.) Therefore, for convenience, the Petition will address the interrelated “priority”-related limitations of [1c], [1d], [1h], and [1i] together. (*Id.*)

For context in interpreting these priority-related limitations, the ’823 specification states only: “An embodiment’s directory update system updates at a higher priority than the documents to be synchronized.” (’823, 9:28-29; EX1015 at 17; Balakrishnan, ¶166.) No affirmative assignment of priority values is described. Nor does the specification explain which component, if any, controls the “higher priority.” (*Id.*) Consistent with the specification’s non-limiting disclosure, claim [1c] does not recite that “the first metadata being assigned a first priority” must be assigned its priority prior to the server system receiving that “first metadata.” (*Id.*) On the contrary, claim [1i] provides that the server may be the only entity that

Petition for *Inter Partes* Review of
U.S. Patent No. 10,754,823

assigns “the” first priority to the first metadata, which could naturally occur **after** the server receives that metadata. (*See* [1i] (“at least one of the server system or the first client device assigns the first priority to the first metadata”); Balakrishnan, ¶166.) Furthermore, the ’823’s Summary mentions only the server **transmitting** metadata before transmitting the associated file, without disclosing the server **receiving** metadata before the file. (’823, 4:3-18; Balakrishnan, ¶166.) Nevertheless, to the extent [1c] requires assigning the “first priority” to the first metadata **before** the server receives that metadata, the prior art discloses and renders that obvious, as discussed below. (*Id.*)

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

Under a first interpretation, if these limitations are interpreted to encompass a server system configured to receive metadata from a first device, and then automatically (not user-manually) transfer that metadata to a second client device

before transmitting associated files, **Sigurdsson** discloses and renders them obvious (in addition to rendering them obvious with Shappell and Rao). (*Id.*, ¶168.)

Sigurdsson discloses and renders obvious that Server 106 receives from the **first client device** a file's generated metadata *before* receiving the file, and then the server automatically transfers that metadata to a **second client device** *before* transmitting the file. (*Id.*, ¶169.) Specifically, Sigurdsson discloses an optional variation that prioritizes transferring metadata prior to files. (*Id.*, ¶¶169-170.) A “priority algorithm” (discussed below), optionally implemented on the clients and/or Server 106, prioritizes the order in which files are transferred. (Sigurdsson, ¶¶58-60, 72; Balakrishnan, ¶170.) Higher-priority (e.g., smaller and/or newer) files may be transferred before lower-priority (e.g., larger and/or older) files. (*See id.*) As a result of the priority algorithm, Server 106 can first receive from the first client device the *metadata* associated with a low-priority file that Server 106 has not received: “The content of this *file may not be transmitted to the Server 106 because of the execution of a priority algorithm* (as discussed earlier). *Instead, metadata describing the file and the time it was accessed may be transmitted to the Server 106.*” (Sigurdsson, ¶72; Balakrishnan, ¶170.) And although it does not appear to be a requirement of the claim, this “metadata describing the file and the time it was accessed” obviously includes metadata generated from the user modifying the file content (which would change the file's description and last-accessed time), matching

the “metadata describing the content, such as the time the [file] was saved” generated when a user modifies a file. (Sigurdsson, ¶34; Balakrishnan, ¶170.)

That metadata received by Server 106 is then transmitted by Server 106 to a **second client device** before Server 106 transfers the associated file. (Balakrishnan, ¶172.) Specifically, that received “*metadata* describing the file and the time it was accessed . . . *may be included in a list tracking content* that was generated” during a period of time when another client (e.g., Client C) was offline. (Sigurdsson, ¶¶69-73; Balakrishnan, ¶172.) Once that client (e.g., Client C) comes back online, it requests and receives from Server 106 that “list of missing content” including the metadata. (Sigurdsson, Fig. 6 (steps 612, 614, 616), ¶¶69-73; Balakrishnan, ¶172.) That client can then use the list to obtain the missing files, which may be transferred by Server 106, such as if no other sources are available or Server 106 is a better source for obtaining the file. (Sigurdsson, ¶¶74, 79-80, 96; Balakrishnan, ¶172.)

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

Under a second interpretation, if the priority-related limitations of [1c], [1d], [1h], and [1i] are interpreted to require affirmative, categorical assignment of greater/lesser priorities to metadata and files, they are additionally rendered obvious

Petition for *Inter Partes* Review of
U.S. Patent No. 10,754,823

by Sigurdsson with Shappell, which teaches prioritizing transfer of metadata for a modified file before transferring the file itself, and Rao, which teaches categorical assignment of first and second “phases” to transfer metadata before files between clients and server. (*Id.*, ¶174.)

As noted, Sigurdsson’s overall “system **400**” (including Server 106 and/or client devices) optionally assigns “priority” to items using a “priority algorithm,” to address the concern that limited available network bandwidth may cause “content sharing [to] fall behind demand.” (Sigurdsson, ¶58; Balakrishnan, ¶175.) For example, “[a]lthough the Server 106 can receive content for all three data items in the order that the user accessed them, *the Server 106 can also deal with the data items by assigning it priorities. The Server 106 can handle the higher priority data items first, and save the lower priority data items for later.*” (Sigurdsson, ¶¶58-59 (describing priority scores assigned to documents/files based on attributes such as “size, type or age”); Balakrishnan, ¶175.) “In some implementations, the priority algorithm is used by *each of the clients* to determine when to transmit content from that client to the server or another client.” (Sigurdsson, ¶58.)

While Sigurdsson does not expressly describe assigning priority values to metadata, Sigurdsson in view of Shappell and Rao renders obvious a predictable variation where the priority algorithm (**priority assignment configuration**) in the system, configured on the first client and/or Server 106, categorically assigns greater

priority to metadata and lower priority to files, thereby causing metadata to transfer prior to the transfer of the associated files based on the greater priority, to advantageously inform clients about updated files. (Balakrishnan, ¶176.)

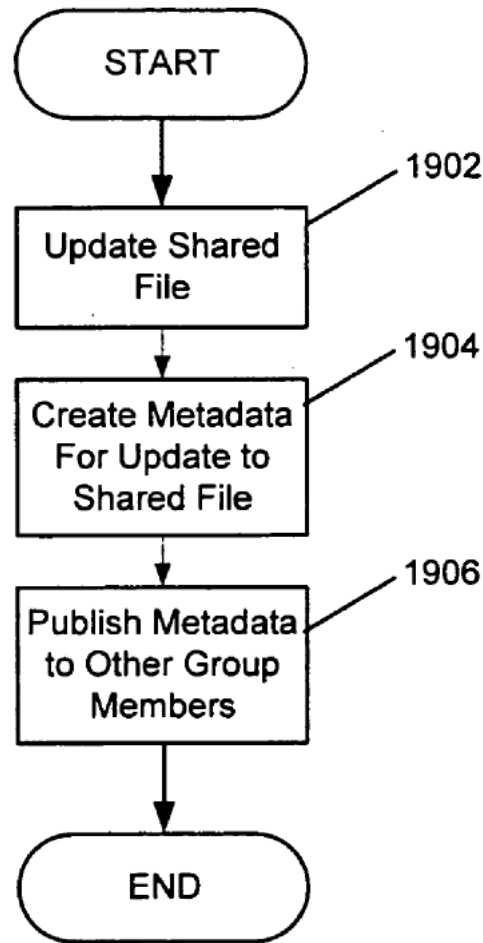
Sigurdsson itself motivates this modification. (Balakrishnan, ¶177.) The purpose of Sigurdsson’s priority algorithm is to prioritize “data items” based on, for example, “size, type or age,” in view of potential limited bandwidth, as “content sharing can fall behind demand.” (Sigurdsson, ¶58; Balakrishnan, ¶177.) Sigurdsson teaches assigning a greater priority to *smaller* and *newer* data items—precisely the attributes a POSA would recognize as applicable to *metadata* (typically containing much less data than the file itself) for an *updated* file—as well as the *type* of “data item,” which obviously would differentiate metadata from file contents. (Sigurdsson, ¶¶58-59; Balakrishnan, ¶177.) A POSA would also appreciate the natural harmony between the priority algorithm and Server 106 prioritizing the “list of missing content” (containing metadata) to inform clients about new content at Server 106, plus Sigurdsson’s teaching that “Server 106 can *handle the higher priority data items first, and save the lower priority data items for later.*” (Sigurdsson, ¶58; Balakrishnan, ¶177.) From these teachings, a POSA would have found it obvious to simply configure the priority algorithm to assign greater priority to metadata, and lower priority to the associated files, advantageously and predictably causing transfer of metadata before the associated files, accounting for

Petition for *Inter Partes* Review of
U.S. Patent No. 10,754,823

Sigurdsson’s “bandwidth” considerations in transferring files. (Sigurdsson, ¶¶60, 79; Balakrishnan, ¶177.) *Boston Sci.*, 554 F.3d at 991 (obvious to combine two adjacent prior art embodiments); *B/E Aerospace v. C&D Zodiac*, 962 F.3d 1373, 1379-80 (Fed. Cir. 2020) (affirming obviousness of “predictable” variation).

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

Like Sigurdsson, **Shappell** teaches synchronizing files across multiple computers using metadata. (Shappell, ¶9; Balakrishnan, ¶178.) Shappell discloses “shared spaces” in which files may be shared among group members’ computers. (Shappell, ¶¶31, 42, 75-81; Balakrishnan, ¶178.) Shappell teaches that “[w]hen a *modification of a file*” occurs, “*metadata is transmitted to all members*” of the group. (Shappell, ¶¶81, 76, Fig. 19 (steps 1902 (“Update Shared File”), 1904 (“Create Metadata For Update to Shared File”), 1906 (“Publish Metadata to Other Group Members”)); Balakrishnan, ¶178.)



(Shappell, Fig. 19.)

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

transparent” icon. (Shappell, ¶¶79, 81; Balakrishnan, ¶178.)

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

Thus, Shappell teaches that a file’s metadata is transferred first (with a higher priority) from an originating computer to all connected group computers *before* transferring a copy of the file itself, to beneficially facilitate user-interface display of the file’s availability. (Balakrishnan, ¶179.)

Rao provides naturally complementary teachings to categorically prioritize synchronizing metadata before files between server and client, to advantageously enable functions at the client. (Balakrishnan, ¶180.) Like Sigurdsson, Rao describes file synchronization between a client and a server over a network. (Rao, ¶¶26, 37; Balakrishnan, ¶180.) Rao discloses synchronizing files between server and client in *two phases*: first the metadata, and then the “raw data” of the files themselves. (Rao, ¶63 (“In the *first phase, metadata is synchronized between a server and client* while the raw data remains on the server system. ... *The second phase of synchronization is to synchronize the selected raw data to the client system.*”); Balakrishnan, ¶180.)

Much like Shappell, Rao teaches that prioritizing metadata enables functions using the metadata beneficial to “the user of the client” receiving the metadata, such as to “display” information about files. (Rao, ¶¶63 (“*the client system . . . may use and manipulate the metadata for any purpose whatsoever*”), 43 (metadata enables “certain functions and operations” for the client system and user, such as “display” information); Balakrishnan, ¶180.)

Motivation to Combine Sigurdsson with Shappell and Rao

It would have been obvious to combine Sigurdsson with Shappell’s and Rao’s metadata-priority teachings, predictably resulting in Sigurdsson’s system where a priority algorithm at the first client and at Server 106 assigns a first, greater priority to metadata and lesser priority to files, resulting in the metadata transferring first (based on the first priority) before transferring associated files. (Balakrishnan, ¶181.)

Sigurdsson, Shappell, and Rao are analogous references in the same field as the ’823, directed to synchronizing files and associated metadata. (Balakrishnan, ¶182.) The combination would have straightforwardly combined prior art elements according to known techniques to predictably improve Sigurdsson’s system in the same way the relevant features enhance Shappell’s and Rao’s disclosed systems, automatically transmitting metadata as a priority based on file changes to timely notify users of an updated version of a file and enable functions (e.g., displaying

information about a modified file) at the receiving client. (Balakrishnan, ¶183.)

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

Furthermore, a POSA would have been motivated to consider additional ways to address Sigurdsson’s stated bandwidth concerns. (Sigurdsson, ¶¶51, 58 (with limited bandwidth, “content sharing can fall behind demand”), 60; Balakrishnan, ¶185.) A POSA would have recognized that Sigurdsson’s system gives rise to a potential gap in synchronicity between versions of a synchronizing “very large document” over a slow network connection, without notifying the user. (Sigurdsson, ¶58; Balakrishnan, ¶185.) Indeed, as Rao cautions, “synchronizing” a very large set of files “may take many minutes or even several hours, depending on the throughput of the communication path . . . among other things.” (Rao, ¶48; Balakrishnan, ¶185.) As an illustration, the user of one client device (e.g., work computer) in Sigurdsson may work on modifying a large file. (Balakrishnan, ¶185.) When the user subsequently accesses another client device (for example, home computer) to continue work on that file, the user would benefit from being able to quickly and visually determine whether or not the version of the file on the home computer is current, before opening the file to work on it. (Balakrishnan, ¶185.) In view of this

potential deficiency in Sigurdsson, a POSA would have been motivated to consider Shappell and Rao for a solution that prioritizes transmitting metadata generated by modifying the file to the user's other client devices (including through Server 106, using its priority algorithm, to account for other client devices that may be currently offline), advantageously leveraging the metadata for "any purpose" such as to "*inform a user*" that an updated file is available. (Shappell, ¶79 (motivating use of metadata "[t]o *inform a user* of a group shared space file that is not present locally" by displaying an indicator "*to denote a 'missing' file that is available on another system*"); Rao, ¶¶63 (motivating that "*the client system . . . may use and manipulate the metadata for any purpose whatsoever*"), 43 (motivating that the metadata enables that "*certain functions and operations may be available to the user*" of the receiving client, including to "display" information regarding files); Balakrishnan, ¶185.)

More generally, the benefits of prioritizing transferring metadata before associated content would have been background knowledge of a POSA from well-known commercially-available software. (Balakrishnan, ¶186.) For example, Microsoft Outlook downloaded headers for email messages (comprising metadata such as subject, date, sender and recipients) before downloading the messages themselves. (Boyce, 366-370; Balakrishnan, ¶186.) Boyce echoes the user interface benefits noted above, explaining an "advantage" of this functionality, "the *ability to*

work with the messages headers of waiting messages without having to download the messages themselves.” (Boyce, 366; Balakrishnan, ¶186.) Echoing Sigurdsson’s bandwidth/priority considerations, Boyce advises that prioritizing metadata/headers “is extremely useful when you’re pressed for time but have a message with a large attachment waiting on the server. You might want to retrieve only your most critical messages without spending the time or connect charges to download that message and its attachment.” (Boyce, 366 (“leaving the one with the large attachment on the server for a less busy time when you can download it”); *id.*, 196 (optionally download headers for messages larger than a specified size); Balakrishnan, ¶186.)

Newsgroup messages in Microsoft Outlook Express included similarly functionality. (Boyce, 307-308; Balakrishnan, ¶187.) Outlook Express would download newsgroup message headers (e.g., subject, size, and date) before the messages themselves, enabling the user to select messages for download. (Boyce, 307-308; Balakrishnan, ¶187.)

A POSA would have appreciated that these advantages of prioritizing metadata for email and newsgroup messages—as well-known ways of sharing files/content—would have similarly applied to the Sigurdsson, Shappell, and Rao file sharing/synchronization systems. (Balakrishnan, ¶188.)

A POSA would have reasonably expected success in implementing this

modified Sigurdsson system, as Sigurdsson already discloses metadata and file transfer using standard, familiar computer, OS, and network components. (Balakrishnan, ¶189.) A POSA would not have been dissuaded by Shappell's primary embodiment being peer-to-peer rather than client-server, as Shappell's relevant features are independent of these types of network architectures. (Balakrishnan, ¶189.) While Shappell's exemplary embodiment is "server-less" (e.g., Shappell, Abstract, ¶7), Shappell does not specifically disparage client/server implementations, proclaims that the "type of network . . . is not critical" and is only "preferably" peer-to-peer, and suggests also transferring "indirectly" between machines such as a client/server architecture would provide. (Shappell, ¶40; Balakrishnan, ¶189.) Similarly, a POSA would not have been dissuaded by Shappell's statement that, 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 teachings to improve Sigurdsson's system. (Balakrishnan, ¶¶189-194.)

(e) [1d]

Sigurdsson with Shappell and Rao render obvious [1d] as explained for [1c] and further below. (Balakrishnan, ¶¶195-196.) As explained for [1c], the references render obvious Sigurdsson's server system **automatically transferring the first metadata over a network to a second client device before transferring the first**

Petition for *Inter Partes* Review of
U.S. Patent No. 10,754,823
file to that device. (Balakrishnan, ¶195.)

As explained above for [1b], Sigurdsson teaches multiple clients (e.g., Clients A-C) **associated with the same user** and communicating with Server 106 **over a network.** (Sigurdsson, ¶¶27-29, 55, 58, Fig. 4; Balakrishnan, ¶196.)

(f) [1e]-[1f]

Sigurdsson in view of Shappell discloses and renders obvious [1e]-[1f]. (Balakrishnan, ¶¶197-211.) To start, as explained for [1c] and [1d], Sigurdsson in view of Shappell and Rao disclose and renders obvious that Server 106 **transfers the first metadata to the second client device before the copy of the first file.** (Balakrishnan, ¶197.) Sigurdsson discloses that its clients may have GUIs, but provides no detail. (Sigurdsson, ¶108; Balakrishnan, ¶198.) A POSA would have been motivated to look to references like Shappell that describe user-interface aspects of Microsoft's Windows operating system to visually indicate file synchronization status to users via GUIs. (*Id.*)

Shappell discloses a “**graphical availability indication**” that, “**before the copy of the first file is transferred to the second client device,**” is “**presented at the second client device based on the first metadata.**” (Balakrishnan, ¶199.) As discussed for [1c], Shappell teaches that a computer's receipt of metadata caused by a file modification causes the display of a “ghosted or semi-transparent” icon. (Shappell, ¶¶79, 81, Figs. 19-20; Balakrishnan, ¶199.) The ghosted icon discloses a

“graphical availability indication” graphically indicating the availability of an
“updated version of the file.” (Shappell, ¶¶79, 81; Balakrishnan, ¶199.)

As explained for [1c], Shappell renders obvious using the **“first metadata”**
described in the foregoing limitations as originating from Client A in Sigurdsson to
cause the presentation of Shappell’s exemplary **“graphical availability indication”**
at Client C (**“second client device”**) based on the **“transfer of the first metadata
to the second client device,”** predictably improving Sigurdsson’s system with a
convenient and user-friendly notification displayed at Client C to inform the user
that an updated file is available from the server. (Shappell, ¶¶76, 79, 81;
Balakrishnan, ¶200.)

Shappell further renders obvious the **“graphical availability indication”** **“is
presented proximate a file icon representing the first file on a user interface of
the second client device”** ([1f]). (Balakrishnan, ¶201.) Beyond the exemplary
“graphical availability indication” (“ghosted” icon) discussed above for [1e],
Shappell renders obvious alternative availability indications presented **proximate a
file icon representing the first file.** (*Id.*)

For context, Shappell teaches the use of the well-known Microsoft Windows
OS, motivating presenting its shared space “through a graphical user interface with
the look and feel of existing file system features,” such as “Windows Explorer-style
dialog boxes,” with ***“graphical user interface presentations and controls that are***

included as part of an existing file system, or at least an intuitive extension of existing file system operations.” (Shappell, ¶¶42, 45, 47, Fig. 3 (“Windows Core OS” layer **302**); Balakrishnan, ¶202.) Windows provided numerous graphical displays of information about a file presented **proximate a file icon**. (Balakrishnan, ¶203.) Shappell teaches several examples, and a POSA would have understood from Shappell’s stated GUI goals that those are merely examples. (Balakrishnan, ¶203.)

For example, Shappell describes that to “enable the user to discern what content is new or has been recently altered, *a ‘new’ glyph may be presented on folders and files that are new or have been recently changed.*” (Shappell, ¶81 (“This glyph provides a visual indicator of new content.”); Balakrishnan, ¶204.) A POSA would have understood this “new” glyph presented “on” “*recently* changed” files is and obviously would be presented **proximate the file icon**, as an additional overlay on the icon that may optionally extend beyond the underlying icon’s visible graphic, which may depend on the size/space of the underlying icon. (Balakrishnan, ¶204.)⁵

As another illustration, where a “missing” file is currently unavailable, “a

⁵ A POSA would have understood “proximate” an icon to encompass overlays of distinct graphics, including overlapping at least in part with the underlying icon. (E.g., EX1023 (describing that “window 5113 is displayed *near* the icon 5105” where window 5113 partly overlaps with icon 5105); Balakrishnan, ¶204 n.11.)

display icon 1506 is presented as a ghosted icon that includes a small image of a red ‘X’ *or other suitable visual indicia* that the file is unavailable to the user.” (Shappell, ¶79; Balakrishnan, ¶205.) The red “X” is presented proximate the file icon. (Balakrishnan, ¶205.)

Considered as a whole, Shappell’s teachings of the “new” glyph indicating files that “have been recently changed,” red “X” indicating file *un*availability, and suggestions to consider “other suitable visual indicia” and “intuitive” GUI extensions, would have prompted a POSA to consider analogous icon-proximate indications for the *availability* of a recently modified “missing” file. (Balakrishnan, ¶205.) The “ghosted” icon **1504** is semi-transparent to indicate a “missing” file that has been modified on another computer, but that “ghosted” icon itself would not inform the user whether the “missing” file was “*recently* changed” or not. (Shappell, ¶¶79, 80, 81; Balakrishnan, ¶205.) A POSA would have considered the “new” glyph a trivially obvious design-choice to present proximate the “ghosted” icon for “*recently* changed” “missing” files (for example, using a time threshold to determine recency), leveraging the “modification date” in the metadata. (Shappell, ¶¶78, 80, 81; Balakrishnan, ¶205.) The glyph would intuitively inform the user, using a GUI feature already present in Shappell’s system: a “new” glyph on the icon of “recently changed” downloaded files and a “new” glyph on the “ghosted” icon for recently modified “missing” files available for download, disclosing **a graphical**

availability indication proximate a file icon. (Shappell, ¶¶78, 80, 81; Balakrishnan, ¶205; *Boston Sci.*, 554 F.3d at 991 (obvious to combine two adjacent embodiments); *Paice v. Ford Motor*, 681 F. App'x 904, 917 (Fed. Cir. 2017) (same, affirming Board).)

Shappell's exhortations to exploit and intuitively extend Windows GUI features would have motivated other **icon-proximate indicators of file availability** obviously supplementing or replacing Shappell's "ghosted" icon. (Balakrishnan, ¶206.) Windows provided numerous icon-proximate indicators for consideration. (Balakrishnan, ¶206.) For example, Shappell describes that "the user may right-click an icon" to "present a graphical interface" such as "Context Menu 1700" shown in Figure 17. (Shappell, ¶83, Fig. 17; Balakrishnan, ¶206.) In this familiar Windows feature, a user right-clicks a file icon, prompting display proximate the icon of a menu of options regarding the file. (Balakrishnan, ¶206.) Shappell does not describe what would happen if a user right-clicked a "ghosted" icon indicating file availability, leaving it to the designer to populate the pop-up "Context Menu" menu, and a POSA would have found it obvious that it would **indicate file availability proximate the icon**, such as displaying a "Download" menu option caused by the receipt of the metadata. (Balakrishnan, ¶206.) In fact, another prior art Microsoft file-synchronization patent (Moromisato) (relied upon to illustrate a POSA's knowledge of Windows features) describes precisely such a pop-up "Download"

menu option, proximately indicating availability to “Download” a file.
(Moromisato, 15:36-43; Balakrishnan, ¶206.)

A POSA similarly would have been motivated to exploit Windows Explorer’s “Details” view that “displays columns of information about each file” (Moromisato, 17:14-21), as another obvious preexisting, convenient, and user-friendly Windows feature to graphically indicate, **proximate a representation of the “missing” file**, the significant information of which files are available for download—again, illustrated by Moromisato (*see id.*), complementing Shappell’s own suggestion to use “Windows Explorer-style” displays (Shappell, ¶42). (Balakrishnan, ¶207.)

A POSA also would have considered obvious, as described in Moromisato, that the icon “that represents that file is *provided with an icon overlay*” that “can be similar to *the ‘hand’ overlay 114 (FIG. 1) displayed by Windows XP Pro* file system when a folder is shared over the network.” (Moromisato, 17:3-14, 15:14-22 (where synchronized “*file downloading is deferred to a later time*, that file is represented in the OS file system as a ‘stub’ file that is very similar to a Windows shortcut,” with “the same icon as the target file, but it *may have an overlay of some sort*”), Fig. 1; Balakrishnan, ¶208.) This familiar Windows “hand” overlay for a network-shared resource discloses another obvious **graphical availability indication proximate a file icon** to be presented proximate the icon for a “missing” file available for download. (Balakrishnan, ¶208.) The classic Windows “shortcut” icon overlay

Petition for *Inter Partes* Review of
U.S. Patent No. 10,754,823

would have been another trivially obvious choice to indicate, **proximate the file’s icon**, that the file is available for download. (Moromisato, 15:14-22 (where synchronized “*file downloading is deferred to a later time*, that file is represented in the OS file system as a ‘stub’ file that is *very similar to a Windows shortcut*,” with “the same icon as the target file, but it *may have an overlay of some sort*”); Balakrishnan, ¶208.)

Yet another obviously well-suited, familiar Windows feature to supplement or replace Shappell’s “ghosted” icon was “Infotips”—a pop-up graphic “to display extended information” **proximate a file icon** when mousing over it, again naturally applying Shappell’s motivation to use existing Windows GUI features to provide information about remotely-located files. (Moromisato, 17:22-29 (illustrating “infotips” in Windows for displaying information about “synchronized files”); Balakrishnan, ¶209.)

Another obvious “intuitive” extension, inspired by the *negative-connotation* red “X” overlay for *un*availability of a “missing” file, would have been a *positive-connotation* overlay proximate the “ghosted” icon for *availability* for download, such as a green checkmark (✓). (Balakrishnan, ¶210.)

The motivation to combine Sigurdsson and Shappell above in [1c] applies equally here. (Balakrishnan, ¶211.) Sigurdsson contemplates standard “home” and “work” computers with Microsoft software but does not expressly describe GUI

features. (Sigurdsson, ¶¶27, 42, 45, 58, 60; Balakrishnan, ¶211.) Shappell picks up where Sigurdsson leaves off, motivating a POSA to consider Windows and Windows-extending GUI features to improve upon Sigurdsson with Shappell's teachings to graphically indicate to users the existence and status of modified files, with full expectation of predictable success. (Balakrishnan, ¶211.)

(g) [1g]

Sigurdsson with Shappell discloses and renders obvious [1g]. (Balakrishnan, ¶¶212-215.) As discussed above for [1e]-[1f], Shappell's disclosed and obviously motivated indicator for an available "missing" file **"indicates that the updated version of the first file generated from the user modifying the content of the first file is available to be downloaded."** (Balakrishnan, ¶212.) Shappell's indicator of a "missing" modified file (including as obviously implemented proximate a file icon, per [1f]) represents that the file is **"available"** to be downloaded *from another system* to the computer displaying the indicator. (Shappell, ¶¶79, 81, 62; Balakrishnan, ¶212.) In the obviously-modified Sigurdsson client-server system, that "other system" is Server 106. (Sigurdsson, ¶¶73-74, 79-80 (explaining that Client C can determine that the missing content is available from the Server and download it from the Server); Balakrishnan, ¶212.) For example, where the second client (e.g., Client C) was offline when the modified file transferred from the first client (e.g., Client A) to Server 106, the metadata

transferred to Client C (per [1c]) after its reconnection would cause the graphical indication that the file is available for download from the server. (Sigurdsson, ¶¶73-74, 79-80; *see also* Sigurdsson, ¶96 (even if server does not contain file, a client may query server to obtain it from another client); Balakrishnan, ¶¶213-215.) The same motivations to combine Sigurdsson with Shappell apply as previously. (Balakrishnan, ¶212.)

(h) [1h]-[1i]

As explained for [1c], and below, Sigurdsson with Shappell and Rao discloses and renders obvious [1h]-[1i]. (Balakrishnan, ¶¶216-217.) Sigurdsson’s “first client” and/or Server 106 obviously would implement the claimed **priority assignment configuration** to assign greater priority to metadata than a priority assigned to files, transferring metadata prior to files, as explained for [1c]. (*Id.*)

2. Claim 2.

As discussed above for [1e]-[1f], Shappell renders obvious modifying Sigurdsson so that, **before the copy of the first file is transferred to the second client device, the transfer of the first metadata causes a file representation (icon) on the user interface of the second device to be updated based on the first metadata.** (Balakrishnan, ¶¶218-219.) As discussed for [1e]-[1f], Shappell teaches that the receipt of metadata from another computer’s “modification of a file” “will *change the visual representation of the file* to a ‘missing’ file” available for

Petition for *Inter Partes* Review of
U.S. Patent No. 10,754,823

download. (Shappell, ¶81; Balakrishnan, ¶219.) All of the obvious GUI design options for presenting the indicator proximate the icon discussed for [1e]-[1f] apply equally here, each updating the file representation (icon). (*Id.*)

3. Claim 3.

(a) [3pre]-[3a]

Sigurdsson discloses and renders obvious [3pre]-[3a]. (Balakrishnan, ¶¶220-225.) Sigurdsson’s clients generally “contain similar components and can accept and transmit content in a similar manner.” (Sigurdsson, ¶¶50-51, 26-28 (user may work on a given file, switching between home and work computers); Balakrishnan, ¶221.) Clients A and C can both similarly post/push files to Server 106. (Sigurdsson, ¶51 (“In the system 400, content still can be shared among the Client A 102, the Client B 104 and a Client C 406 using the Server 106 in a method similar to that of method 200 of FIG. 2.”); ¶32 (“FIG. 2 is a sequence diagram showing an illustrative method 200 for sharing and synchronizing content across multiple client devices according to the implementation shown in FIG. 1.”); Balakrishnan, ¶221.) Thus, as explained for [1b], Sigurdsson renders obvious [3pre]-[3a] regarding Client C (“**second client device**”). (Balakrishnan, ¶221.) As one exemplary “**second file**,” Sigurdsson discloses an “office memo on the Client C.” (Sigurdsson, ¶58; Balakrishnan, ¶221.)

(b) [3b]

Sigurdsson discloses: “*When a client comes online, it can notify the Server*

106 that it is online, and the Server 106 can update its table of clients with the online status.” (Sigurdsson, ¶61; Balakrishnan, ¶222.) This notification from client to server allows the server to “**determine that the server system is in communication with the first client device associated with the user.**” (Balakrishnan, ¶222.) For the reasons above regarding [1b] and [3a], Sigurdsson’s clients all operate similarly, and it would have been obvious to determine that Client A (“**first client device**”), or any other client, is in communication with the server. (Balakrishnan, ¶¶222-223.)

(c) [3c]

As noted above, Sigurdsson’s clients are fungible, using the same components to implement the same synchronization functionality. (Sigurdsson, ¶¶50-51; Balakrishnan, ¶224.) Thus, it would have been obvious to a POSA that the functionality described above for claim 1 with respect to Clients A and C, further described in paragraphs 72-74 and 79-80, would be performed in the **reverse direction**. (Sigurdsson, ¶¶72-74, 79-80; Balakrishnan, ¶224.) Client C would send the “**second file**” to the server, which, in turn, would determine that it is in communication with Client A (after Client A has been offline, for example). (Balakrishnan, ¶224.) Thereafter, the server would send Client A the “list identifying the missing content” referenced in paragraph 73 and, as described in paragraph 79-80, send missing content (i.e., the “**second file**”) to Client A. (Sigurdsson, ¶¶73, 79-80; Balakrishnan, ¶224.)

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

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

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 ensure both are updated.

(Balakrishnan, ¶224.) For example, Sigurdsson discusses a user's work and home computers as two exemplary "clients," and describes example use cases such as a user editing a Word document, synchronized between those computers. (Sigurdsson, ¶¶26-28 (further describing Server 106 as optional "drop box" "conduit" between client devices); Balakrishnan, ¶224.)

A POSA would have found it obvious that in such a use case, after the copy of the Word document (which may be called "Version 1") is transmitted from the user's home computer to work computer, the user may continue editing it at work. (Balakrishnan, ¶224.) 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. (Balakrishnan, ¶224.) However, Sigurdsson does not detail what would happen at the user's home computer in that circumstance. (Balakrishnan, ¶224.) A POSA would have been motivated to look for solutions to address outdated versions of files (like "Version 1"), and would have found it obvious that one possibility would be for the updated modified file ("Version 2") to **replace** the outdated older version ("Version 1") on the home computer, for reasons including the obvious benefit of preventing copies of outdated versions from endlessly accumulating in storage on the user's computers. (Balakrishnan, ¶224.) Furthermore, Shappell expressly teaches and motivates this

feature with its “automatic replication” option, as discussed further below.

Shappell’s “automatic replication” option further renders obvious implementing Sigurdsson’s Server 106 to **automatically transfer the copy of the second file to the first client device to replace an older version of the file responsive to determining that the server is in communication with the first client device and receiving the copy of the second file from the second client device.** (Balakrishnan, ¶224.) Shappell teaches a user-selectable “File Replication” setting where the user can switch between (1) manually downloading files “on an on-demand basis,” and (2) automatically downloading files when available, with an “automatic replication” setting. (Shappell, ¶¶62, 81; Balakrishnan, ¶224.) With “automatic replication,” an “out-of-date file” is “remove[d]” from the receiving computer and replaced with an “updated version.” (Shappell, ¶81 (“When a modification of a file contained in a shared space occurs, a notification to other members in the group occurs (see steps 1902, 1904 and 1906 in FIG. 19). In a preferred embodiment, upon receipt of such notification, member machines that have previously obtained a copy of the shared file will *remove the local copy of the out-of-date file. . . . [I]f automatic replication of the shared file is enabled, the local machine obtains an updated version of the file*, as is shown in step 2006 in FIG. 20.”), Figs. 19-20; Balakrishnan, ¶224.)

It would have been obvious to implement Shappell’s “automatic replication”

feature as an option for each Sigurdsson client device, including Client A (“**first client device**”), so that Server 106 would **automatically** transfer to Client A the copy of a file modified at Client C (the “**second client device**”), to ensure the user at Client A has the updated file. (Balakrishnan, ¶225.) A POSA would have been motivated to incorporate Shappell’s auto-replication option to address Sigurdsson’s concern for each client “to keep its locally stored content synchronized with content from the other clients,” and it easily and obviously would be implemented applying Sigurdsson’s teaching that each client can “maintain an open connection” with the server. (Sigurdsson, ¶¶84-85; Balakrishnan, ¶225.) The motivations to combine Sigurdsson with Shappell and Rao for [1c] further apply here as well. (Balakrishnan, ¶¶181-194.)

4. Claim 4.

For the reasons above for claim 1 and below, Sigurdsson with Rao and Shappell discloses and renders obvious claim 4. (Balakrishnan, ¶¶226-227.) Sigurdsson discloses that Client A “posts” the copy of the file, which Sigurdsson also refers to as a “push,” responsive to the user modifying the content of the file: “The Client A 102 *posts* Client A Content 108 to the Server 106 *when content is changed on* the Client A 102. . . . In this connection-oriented architecture . . . , each of the clients can *push* new data” (Sigurdsson, ¶84; *id.*, ¶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, ¶227.) A POSA would also understand that HTTP POST discloses a “push request.” (Balakrishnan, ¶227.) It would have been obvious to incorporate this functionality associated with Figure 7 from Sigurdsson into other embodiments in Sigurdsson, such as the embodiment shown in Figure 6 and discussed above, given that they reference the same components (Server 106 and Clients A, B, and C) and given that the clients are described as being fungible and used for similar purposes, as discussed above. (Sigurdsson, ¶¶50-51; Balakrishnan, ¶227; *Boston Sci.*, 554 F.3d at 991 (obvious to combine adjacent embodiments).)

5. Claim 8.

Claim 1 recites a system that executes the same method steps of claim 8, which is obvious for the same reasons. (Balakrishnan, ¶¶228-229.)

6. Claims 9-11.

Claims 9-11 are substantively identical to claims 2-4, respectively, and are obvious for the same reasons. (Balakrishnan, ¶¶230-235.)

7. Claim 13.

Claim 13 is substantively identical to claim 1, except it adds “[o]ne or more **non-transitory machine-readable media storing instructions that, when executed by one or more processors of a server system, cause operations.**” Sigurdsson discloses that Server 106 can include “System 900” in Figure 9. (Sigurdsson, ¶104; Balakrishnan, ¶236.) System 900 includes “a **Memory 920**” and

Petition for *Inter Partes* Review of
U.S. Patent No. 10,754,823

a “***Storage Device 930.***” (Sigurdsson, ¶105; Balakrishnan, ¶237.) Memory **920** may be a “non-volatile memory unit” and “[s]torage devices suitable for tangibly embodying computer program instructions and data include all forms of non-volatile memory.” (Sigurdsson, ¶¶106, 110; Balakrishnan, ¶¶237-239.) Thus, this claim would have been obvious over Sigurdsson, Shappell, and Rao for these reasons and those above for claim 1. (*Id.*)

8. Claim 14-16.

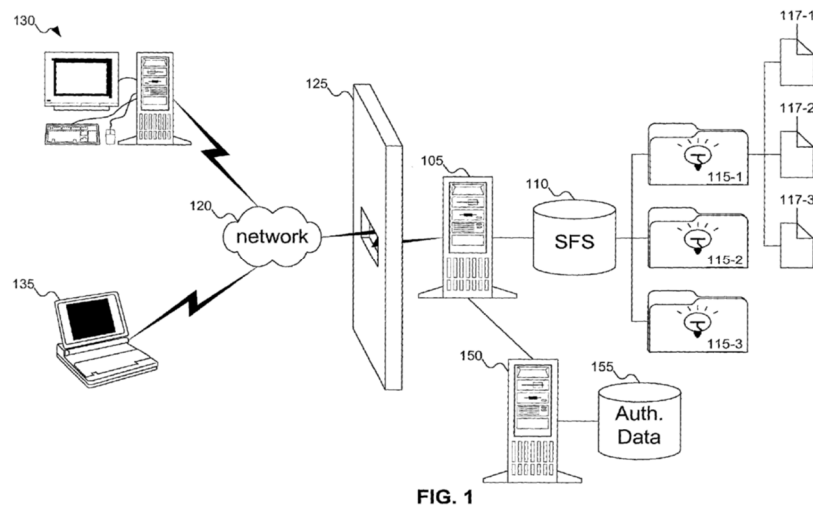
Claims 14-16 are substantively identical to claims 2-4, respectively, and obvious for the same reasons. (Balakrishnan, ¶¶240-245.)

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

1. Claim 1.

(a) [1pre]

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



(b) [1a]

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

(c) [1b]

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

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

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

Brown describes that the **user modifies a content of an electronic file stored**

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

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

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

Petition for *Inter Partes* Review of
U.S. Patent No. 10,754,823

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

Motivation to Combine Brown with Hesselink

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

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

Brown and Hesselink are analogous references in the same field as the '823, directed to synchronizing files across a user's devices. (Balakrishnan, ¶602.) The combination would have straightforwardly combined prior art elements according to known techniques to yield a predictable improvement to Brown's system, by synchronizing files from a client device to Brown's server responsive to the user modifying the file. (Balakrishnan, ¶603.)

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

concern that a user is not using the latest, most up to date version of a file.”

(Hesselink, ¶189; Balakrishnan, ¶605.)

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

and contemplates that each synchronized computer can be a “file server.”
(Hesselink, ¶¶23-34, 122; Balakrishnan, ¶611.)

(d) [1c]

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

Brown alone discloses and renders obvious (and renders obvious with Shappell and Rao) the priority-related limitations of [1c], [1d], [1h], and [1i] at least to the same extent the ’823 discloses them. (*Id.*) Brown’s server **receives over a network from the first client device** (e.g., computer 130) **metadata associated with the modified file that is generated from the user modifying the content of the file**, and Brown discloses and renders obvious that this metadata associated with a modified file is **transferred with greater priority than the file itself** between the client and server. (Balakrishnan, ¶613.)

Brown describes various **metadata**, including update time, file length/size, IDs, and other data generated from “file system activity,” which includes the user modifying file content. (Brown, 5:25-6:10, 6:61-7:5, 7:21-43; Balakrishnan, ¶¶613-616.) During file synchronization, the client transfers updated metadata to the server. (Balakrishnan, ¶614.) Brown specifically describes that **metadata** for a modified file is uploaded from the originating “**first client device**” to the server (e.g.,

“CSI,” *see* Brown, 10:9-10, “SID” (“Server ID”), “parent directory SID,” and “file name,” *id.*, 15:38-40, 5:50-52) prior to the modified file being uploaded. (Brown, 12:53-13:3 (upload SID prior to modified file), 13:57-62 and Fig. 8A (step **805**, upload CSI prior to modified file), 14:5-6 and Fig. 8B (step **850**, upload changes, detailed at Figs. 11A-C), Figs. 11A-C and 15:30-54 (step **1115** upload SID, parent directory SID, file name; step **1120** upload modified file); Balakrishnan, ¶¶614-619.) Accordingly, the system is configured so that the originating client **sends the first metadata to the server prior to sending the file associated with such metadata to the server.** (*Id.*)

Then, **before** the server transfers a copy of the modified file to a “**second client device**” (another synchronizing client, e.g., computer **135**), the server first transfers to that receiving client the Server Synchronization Data (“SSD”) that contains the **metadata** of all files “*that have been changed*” since the last synch by the client. (Brown, 5:25-49, 10:19-27, 8:48-61 (describing SSD contents including SID), 2:65-67 (“Client machines synchronize with each other by synchronizing to a common server account.”); Balakrishnan, ¶¶616, 681.) Figures 8B and 9A show the server-to-client synchronization process including (1) step **825**, client receives SSD (including the file metadata) from the server, followed by (2) step **915**, client downloads the file from the server, which is further described in Figures 10A-10B for a modified file, including step **1050** downloading an “entire” modified file from

Petition for *Inter Partes* Review of
U.S. Patent No. 10,754,823

the server. (Brown, Figs. 8B (step **825**), 9A (step **915**), 10A-10B (step **1050**), 13:65-14:3, 14:7-19, 14:60-15:16; Balakrishnan, ¶¶616-619.)

To the extent the “priority”-related limitations of [1c], [1d], and [1i] are interpreted to require further categorical assignment of priorities to metadata and files, Brown in view of Shappell and Rao discloses and renders them obvious for similar reasons as Ground 1 above. (Balakrishnan, ¶622.)

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

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

Brown itself supports this modification. (Balakrishnan, ¶¶623-645.) As discussed above, Brown streamlines data exchanged with the server by comparing the number of modified portions of a file with a threshold level to determine whether

Petition for *Inter Partes* Review of
U.S. Patent No. 10,754,823

to transmit the entire file or only modified portions. (Brown, 11:53-13:40; Balakrishnan, ¶684.) Brown’s teachings would have motivated a POSA to consider solutions that streamline and prioritize data transmissions. (Balakrishnan, ¶¶623-645.)

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

As explained for Ground 1, **Rao** provides complementary teachings to categorically prioritize synchronizing metadata before files in two “phases” between server and client, to advantageously enable functions at the client, which would be naturally applicable to Brown’s client/server synchronization system. (Rao, ¶¶26, 37, 43, 63; Balakrishnan, ¶631.)

Motivation to Combine Brown, Hesselink, Shappell, and Rao

Brown, Hesselink, Shappell, and Rao are analogous references in the same field as the ’823, directed to synchronizing files and associated metadata. (Balakrishnan, ¶¶632-645.) The combination would have straightforwardly

combined prior art elements according to known techniques to predictably improve the Brown/Hesselink system in the same way the relevant features enhance Shappell's and Rao's disclosed systems, by automatically transmitting metadata as a priority based on file changes to timely notify users of an updated version of a file and enable functions (e.g., displaying information about a modified file) at the receiving client. (Balakrishnan, ¶634.)

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

Significantly, even where Brown's system is modified in view of Hesselink so that each client transfers a copy of a modified file to the server as the user modifies the file (where Brown's file-changes-threshold is exceeded), another client device might not receive a copy of the modified file until after a substantial amount of time

has passed, leaving the user uninformed about the modification, such as where a relatively large file is being transferred over a slow connection. Brown explains: “*if a user has a slow network connection . . . it can be time-consuming to have to wait for the entire document to upload or download.*” (Brown, 12:8-12; Balakrishnan, ¶624.) Rao confirms that large file transfers could take “many minutes or even several hours.” (Rao, ¶48; Balakrishnan, ¶636.) A POSA would have recognized that Brown’s system thus gives rise to a potential gap in synchronicity between versions of a synchronizing document. (Balakrishnan, ¶636.) For example, the user of a synchronizing client device (e.g., work computer) in Brown may work on modifying the large file. (*Id.*) When the user subsequently accesses another synchronizing client device (for example, home computer) to continue work on that file, the user would benefit from being able to quickly and visually determine whether or not the version of the file on the home computer is current, before opening the file to work on it. (*Id.*)

A POSA would have looked to references like Shappell and Rao to address this issue, predictably resulting in the Brown/Hesselink system with a priority assignment configuration where the metadata associated with a file modified at the originating first client device (for example, device **130**) is transmitted automatically by that device to the server, and then by the server **to a second client device** (for example, device **135**), **with a first priority that is higher than a second priority**

that is assigned to files, thereby enabling a graphical availability indication at the second client device (as discussed further for [1e]-[1f] below), advantageously promptly notifying the user at that device of the updated (but still not yet downloaded) file across the network. (Balakrishnan, ¶636.)

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

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

Hesselink also would not dissuade a POSA from combining the above-discussed features. (Balakrishnan, ¶¶644-645.) Hesselink discusses computers

running Microsoft Windows (Hesselink ¶¶78, 187, 203), a connection server **14** running Windows NT (*id.*, ¶80), and Windows Explorer “to access remote storage” (*id.*, ¶¶127-128, 133-134). (*Id.*, ¶644.) A POSA would have viewed Microsoft references like Shappell and Rao, which discuss Microsoft GUIs, as complementary, particularly as Hesselink discusses Windows Explorer’s use of “file overhead information” (i.e., metadata) “*to list or display directory data structures for user navigation purposes.*” (Hesselink, ¶134; Balakrishnan, ¶644.) Hesselink also discusses optional “subscriptions” for metadata (“file overhead” data) and file data, further supporting a POSA’s consideration of references like Shappell and Rao for the benefits of prioritizing transfer of metadata over file data to facilitate graphical indications of file availability. (Hesselink, ¶136; Balakrishnan, ¶644.)

(e) [1d]

Brown with Shappell and Rao discloses and renders obvious [1d] for the reasons discussed for [1c] above, with Brown’s “**second client device**” (e.g., computer **135**) obviously associated with the same user as explained for [1b]. (Balakrishnan, ¶¶646-647.)

(f) [1e]-[1f]

Brown with Shappell discloses and renders obvious [1e]-[1f]. (Balakrishnan, ¶¶648-663.) As discussed for [1c], Brown with Shappell and Rao discloses and renders obvious that the server automatically transfers file metadata to the

synchronizing client device prior to transferring the copy of the file. (Balakrishnan, ¶649.)

Brown and Hesselink disclose that their clients run operating systems with GUIs, such as Microsoft Windows, but do not detail the GUIs regarding synchronizing files. (Brown, 7:6-11; Hesselink, ¶187; Balakrishnan, ¶650.) Accordingly, a POSA would have been motivated to consider how to indicate to users the current state of files being synchronized using the Brown/Hesselink system and would have been familiar with Microsoft Windows as a well-known operating system including GUI components for indicating the state of file transfers and sharing, as disclosed in Shappell, for example. Thus, a POSA would have looked to references like Shappell for insight into how to visually indicate file synchronization status via GUIs. (*Id.*)

In the modified Brown system incorporating Shappell’s teachings, the graphical availability indication would indicate that the updated version of the file is both available for download from the server and present at the originating client device (for example, desktop computer **130**), after the originating client device uploaded a modified file to the server. (Balakrishnan, ¶667.)

As discussed above for Ground 1, Shappell discloses a **“graphical availability indication”** that, **“before the copy of the first file is transferred to the second client device,”** is **“presented at the second client device based on the**

first metadata” and renders obvious using the **“first metadata”** to cause the presentation of Shappell’s exemplary **“graphical availability indication”** based on the **“transfer of the first metadata to the second client device,”** which could be used to predictably improve Brown’s system with a convenient and user-friendly notification displayed at the receiving client (e.g., portable computer **135**) to inform the user that an updated file is available from the server. (Balakrishnan, ¶¶198-211.)

As discussed above for Ground 1, Shappell with a POSA’s knowledge further renders obvious that the **“graphical availability indication” “is presented proximate a file icon representing the first file on a user interface of the second client device”** ([1f]). The motivation to combine Brown and Shappell above in [1c] applies with full force here. (Balakrishnan, ¶¶201-211.) Brown contemplates standard “home” and “work” computers with Microsoft software, but does not expressly describe GUI features. (Brown, 1:25-29, 7:6-11; Balakrishnan, ¶¶650, 663.) Shappell picks up where Brown leaves off, motivating a POSA to consider Windows and Windows-extending GUI features to improve upon Brown with Shappell’s teachings to graphically indicate to Brown system users the existence and status of modified files, with full expectation of predictable success. (Balakrishnan, ¶663.)

(g) [1g]

Brown with Shappell discloses and renders obvious [1g]. (Balakrishnan,

¶¶664-667.) As discussed above for [1e]-[1f], Brown with Hesselink discloses operating systems having GUIs and functionality for sending updated files from one client to a server and, in turn, from the server to another client. (*Id.*, ¶¶648-663.) In addition, as discussed above for [1e]-[1f], it would have been obvious to implement GUI indications that updated versions of files are available for download. (*Id.*) Shappell's indicator of a "missing" modified file (including as obviously implemented proximate a file icon, per [1f]) represents that the file is "***available on another system***" to be downloaded to the computer displaying the indicator. (Shappell, ¶¶79, 81, 62; Balakrishnan, ¶666.) In the obviously-modified Brown client-server system, the "other system" would be the server, with the display at the "second" client device (e.g., computer **135**) indicating that the file is available for download from the server. (Brown, 4:19-24; Balakrishnan, ¶¶666-667.) The same motivations to combine Brown with Shappell apply as previously. (Balakrishnan, ¶¶601-611, 632-645, 666.)

(h) [1h]-[1i]

As explained for [1c], Brown with Shappell and Rao discloses and renders obvious [1h]-[1i]. (Balakrishnan, ¶¶668-671.)

2. Claim 2.

As discussed for [1e]-[1f], Shappell renders obvious modifying Brown's system so that, **before the copy of the first file is transferred to the second client**

device, the transfer of the first metadata causes a file representation on the user interface of the second device to be updated based on the first metadata – specifically, receiving metadata from another computer’s “modification of a file” “*change[s] the visual representation of the file* to a ‘missing’ file” “ghosted” or semi-transparent icon. (Shappell, ¶¶79, 81.) The obvious GUI design-options discussed for [1e]-[1f] apply equally here, updating the file representation (icon). (Balakrishnan, ¶¶672-673.)

3. Claim 3.

(a) [3pre]-[3a]

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

(b) [3b]

Brown with Hesselink discloses and renders obvious [3b]. (Balakrishnan, ¶¶676-678.) For example, it would have been obvious that when synchronizing the server with each Brown client device, the client would initiate communication with the server, and the server would respond based on a determination that client and server are in communication, and obviously using well-known handshake type functions in TCP/IP or HTTP or other protocols, as Brown teaches using TCP/IP and HTTP (or “other protocols”): “The client SA communicates with the server SFS via TCP/IP. Preferably, the client SA communicates with the server SFS using a proprietary protocol tunneled within the hypertext transport protocol (HTTP), but a person skilled in the art will recognize that other protocols can be used. *Communication between the client SA and the server SFS is initiated by the client SA and responded to by the server SFS.*” (Brown, 3:1-8; Balakrishnan, ¶677.)

Hesselink further renders obvious this element, disclosing determining whether computers are online/offline and, if offline, storing modifications for later transmission, with offline computers “automatically updated as soon as those computers go online again.” (Hesselink, ¶¶183, 197 (after “re-establish a connection,” “updates (including any updates to file data . . .) may be automatically transmitted” by a user module); Balakrishnan, ¶678.) A POSA would have been motivated to consider Hesselink’s teachings to serve Brown’s goal of “*maintaining*

currency at each client as changes are made,” with Brown encouraging “any scheduling algorithm to initiate synchronization.” (Brown, 1:47-50, 3:44-46; Balakrishnan, ¶678.) The same motivations to combine Brown and Hesselink for [1b] apply here. (Balakrishnan, ¶¶601-611, 632-645.)

(c) [3c]

As discussed for [1b]-[1c] and [3pre]-[3a], Brown’s clients are functionally interchangeable, bi-directionally “synchronizing to a common server account.” (Brown, 2:65-67; Balakrishnan, ¶¶679-686.) Brown with Hesselink discloses and renders obvious [3c]. (Balakrishnan, ¶680.) Responsive to Brown’s server determining that it is in communication with the first client device (e.g., computer 130) per [3b], the server transmits the second file received from the second client (e.g., computer 135) per [3a]. (Balakrishnan, ¶¶679-686.)

Brown teaches that the server application (SFS) automatically transfers the modified file received from the second device to the first device to **replace an older version of the file**. (Balakrishnan, ¶683.) As discussed above for [1c], it would have been obvious to update another device associated with the same user by transferring an updated version of a file received from one user device to the other. Brown’s client devices synchronize bi-directionally “with each other by synchronizing to a common server account.” (Brown, 2:65-67; Balakrishnan, ¶683.) After the server account receives a modified file from one client device, the server

Petition for *Inter Partes* Review of
U.S. Patent No. 10,754,823

application transfers that file to another client device associated with the common server account. (Brown, 10:18-27, 10:53-55; Balakrishnan, ¶683.) The downloaded copy of a modified file “**replac[es] an older version of the file.**” (Brown, 13:21-23; Balakrishnan, ¶684.)

Hesselink further renders obvious implementing Brown’s server to automatically transmit modified files to connected clients responsive to (i) determining that it is in communication with clients, and (ii) receiving the modified files. As discussed for [1b], a POSA would have been motivated to consider Hesselink’s automatic-update teachings to serve Brown’s goal of “*maintaining currency at each client as changes are made*” using “any scheduling algorithm to initiate synchronization.” (Brown, 1:47-50, 3:44-46; Balakrishnan, ¶686.) As discussed for [1b] and [3b], Hesselink teaches to “*automatically update*” connected computers with modifications to files, including currently connected computers and computers that “go online again” after being offline. (Hesselink, ¶¶183, 197; Balakrishnan, ¶¶685-686.) Brown’s and Hesselink’s complementary teachings would have motivated a POSA to implement Brown’s server to automatically transmit modified files to connected clients as soon as file updates occur. (Balakrishnan, ¶686.) The same motivations to combine Brown and Hesselink for [1b] apply here. (Balakrishnan, ¶¶601-611, 632-645.)

4. Claim 4.

Brown and Hesselink disclose and render obvious claim 4's limitations. Brown's client Synchronization Applications (SAs) "initiate" communication that the server SFS "respond[s] to." (Brown, 2:59-3:8; Balakrishnan, ¶¶687-692.) Brown's clients submit "*push*" requests to the server. (Brown, 4:62-64 (clients may "push changes to the server"), 15:30-35 ("*push* changes to the server from a client"); *Id.*, Abstract (clients "*push* local changes of the files onto the server."); Balakrishnan, ¶¶691-692.) As explained for [1b], Hesselink renders obvious Brown's clients automatically pushing changed files responsive to user file modifications. (Balakrishnan, ¶¶584-611.)

5. Claim 8.

Claim 1 recites a system that executes the same method steps of claim 8, which is obvious for the same reasons. (Balakrishnan, ¶¶693-694.)

6. Claims 9-11.

Claims 9-11 are substantively identical to claims 2-4, respectively, and are obvious for the same reasons. (Balakrishnan, ¶¶695-700.)

7. Claim 13.

Claim 13 recites "[o]ne or more non-transitory machine-readable media storing instructions that, when executed by one or more processors of a server system, cause operations comprising" the same steps as claim 1. (Balakrishnan, ¶¶701-703.) As discussed above for claim 1, and below, claim 13 would have been

Petition for *Inter Partes* Review of
U.S. Patent No. 10,754,823

obvious. (Balakrishnan, ¶¶701-703.) Brown discloses that its server SFS application is “[t]ypically” “installed on a hard disk drive within server 105” or “other forms of media . . . or optical media.” (Brown, 3:59-62; Balakrishnan ¶703.) A POSA would have understood a hard disk (or other media) includes “[o]ne or more non-transitory machine-readable media storing instructions that, when executed by one or more processors of a server system,” cause the operations on Brown’s server. (*Id.*)

8. Claims 14-16.

Claims 14-16 are substantively identical to claims 2-4, respectively, and obvious for the same reasons. (Balakrishnan, ¶¶704-709.)

IX. CONCLUSION

Petitioners respectfully request IPR institution.

Petition for *Inter Partes* Review of
U.S. Patent No. 10,754,823

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. 10,754,823

APPENDIX A

Claims Listing
Claims 1-4, 8-11, 13-16
U.S. Pat. No. 10,754,823

Claim 1:

[1pre] A system comprising:

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

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

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

[1d] automatically transfer, based on the first priority being greater than the second priority, the first metadata over a network to a second client device associated with the user such that the first metadata is transferred to the second client device before the copy of the first file is transferred to the second client device,

[1e] wherein, before the copy of the first file is transferred to the second client device: (i) the transfer of the first metadata to the second client device causes a graphical availability indication of the updated version of the first file to be presented at the second client device based on the first metadata, and

[1f] (ii) the graphical availability indication is presented proximate a file icon representing the first file on a user interface of the second client device, and

[1g] wherein the graphical availability indication indicates that the updated version of the first file generated from the user modifying the content of the first file is available to be downloaded from the server system to the second client device; and

[1h] subsequent to the transfer of the first metadata to the second client device, transfer the copy of the first file to the second client device,

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

Claim 2: The system of claim 1, wherein, before the copy of the first file is transferred to the second client device, the transfer of the first metadata to the second client device causes a file representation of the first file presented on the user interface of the second client device to be updated based on the first metadata.

Claim 3:

[3pre] The system of claim 1, wherein the computer program instructions, when executed, cause the server system to:

[3a] receive a copy of a second file from the second client device associated with the user, wherein the copy of the second file is automatically received from the second client device responsive to the user modifying a content of the second file stored on the second client device, the copy of the second file being an updated version of the second file that is generated from the user modifying the content of the second file;

[3b] determine that the server system is in communication with the first client device associated with the user; and

[3c] automatically transfer the copy of the second file to the first client device associated with the user to replace an older version of the second file stored on the first client device, responsive to (i) determining that the server system is in communication with the first client device and (ii) receiving the copy of the second file from the second client device.

Claim 4: The system of claim 1, wherein the copy of the first file is automatically received from the first client device responsive to (i) a push request of the first client device and (ii) the user modifying the content of the first file stored on the first client device.

Claim 8:

[8pre] A method being implemented by a server system comprising one or more processors executing computer program instructions that, when executed, perform the method, the method comprising:

[8a] receiving, over a network, a copy of a first file from a first client device associated with a user, wherein the copy of the first file is automatically received from the first client device responsive to the user modifying a content of the first file stored on the first client device, the copy of the first file being an updated version of the first file that is generated from the user modifying the content of the first file;

[8b] receiving, over a network, from the first client device, first metadata associated with the updated version of the first file that is generated from the user modifying the content of the first file, the first metadata being assigned a first priority greater than a second priority assigned to the copy of the first file;

[8c] automatically transferring, based on the first priority being greater than the second priority, the first metadata over a network to a second client device associated with the user such that the first metadata is transferred to the second client device before the copy of the first file is transferred to the second client device,

[8d] wherein, before the copy of the first file is transferred to the second client device: (i) the transfer of the first metadata to the second client device causes a graphical availability indication of the updated version of the first file to be presented at the second client device based on the first metadata, and

[8e] (ii) the graphical availability indication is presented proximate a graphical file representation of the first file on a user interface of the second client device, and

[8f] wherein the graphical availability indication indicates that the updated version of the first file generated from the user modifying the content of the first file is available to be downloaded from the server system to the second client device; and

[8g] subsequent to the transfer of the first metadata to the second client device, transferring the copy of the first file to the second client device,

[8h] wherein at least one of the server system or the first client device comprises a priority assignment configuration to assign greater priority to metadata associated with files than priority assigned to the files such that at least one of the server system or the first client device assigns the first priority to the first metadata and the second priority to the copy of the first file based on the priority assignment configuration.

Claim 9: The method of claim 8, wherein, before the copy of the first file is transferred to the second client device, the transfer of the first metadata to the second client device causes a file representation of the first file presented on the user interface of the second client device to be updated based on the first metadata.

Petition for *Inter Partes* Review of
U.S. Patent No. 10,754,823

Claim 10:

[10pre] The method of claim 8, further comprising:

[10a] receiving a copy of a second file from the second client device associated with the user, wherein the copy of the second file is automatically received from the second client device responsive to the user modifying a content of the second file stored on the second client device, the copy of the second file being an updated version of the second file that is generated from the user modifying the content of the second file;

[10b] determining that the server system is in communication with the first client device associated with the user; and

[10c] automatically transferring the copy of the second file to the first client device associated with the user to replace an older version of the second file stored on the first client device, responsive to (i) determining that the server system is in communication with the first client device and (ii) receiving the copy of the second file from the second client device.

Claim 11: The method of claim 8, wherein the copy of the first file is automatically received from the first client device responsive to (i) a push request of the first client device and (ii) the user modifying the content of the first file stored on the first client device.

Petition for *Inter Partes* Review of
U.S. Patent No. 10,754,823

Claim 13:

[13pre] One or more non-transitory machine-readable media storing instructions that, when executed by one or more processors of a server system, cause operations comprising:

[13a] receiving, over a network, a copy of a first file from a first client device associated with a user, wherein the copy of the first file is automatically received from the first client device responsive to the user modifying a content of the first file stored on the first client device, the copy of the first file being an updated version of the first file that is generated from the user modifying the content of the first file;

[13b] receiving, over a network, from the first client device, first metadata associated with the updated version of the first file that is generated from the user modifying the content of the first file, the first metadata being assigned a first priority greater than a second priority assigned to the copy of the first file;

[13c] automatically transferring, based on the first priority being greater than the second priority, the first metadata over a network to a second client device associated with the user such that the first metadata is transferred to the second client device before the copy of the first file is transferred to the second client device,

[13d] wherein, before the copy of the first file is transferred to the second client device: (i) the transfer of the first metadata to the second client device causes an availability indication of the updated version of the first file to be presented at the second client device based on the first metadata, and

[13e] (ii) the availability indication is presented proximate a graphical file representation of the first file on a user interface of the second client device, and

[13f] wherein the availability indication indicates that the updated version of the first file generated from the user modifying the content of the first file is available to be downloaded from the server system to the second client device; and

[13g] subsequent to the transfer of the first metadata to the second client device, transferring the copy of the first file to the second client device,

[13h] wherein at least one of the server system or the first client device comprises a priority assignment configuration to assign greater priority to metadata associated with files than priority assigned to the files such that at least one of the server system or the first client device assigns the first priority to the first metadata and the second priority to the copy of the first file based on the priority assignment configuration.

Claim 14: The media of claim 13, wherein, before the copy of the first file is transferred to the second client device, the transfer of the first metadata to the second client device causes a file representation of the first file presented on the user interface of the second client device to be updated based on the first metadata.

Claim 15:

[15pre] The media of claim 13, the operations further comprising:

Petition for *Inter Partes* Review of
U.S. Patent No. 10,754,823

[15a] receiving a copy of a second file from the second client device associated with the user, wherein the copy of the second file is automatically received from the second client device responsive to the user modifying a content of the second file stored on the second client device, the copy of the second file being an updated version of the second file that is generated from the user modifying the content of the second file;

[15b] determining that the server system is in communication with the first client device associated with the user; and

[15c] automatically transferring the copy of the second file to the first client device associated with the user to replace an older version of the second file stored on the first client device, responsive to (i) determining that the server system is in communication with the first client device and (ii) receiving the copy of the second file from the second client device.

Claim 16: The media of claim 13, wherein the copy of the first file is automatically received from the first client device responsive to (i) a push request of the first client device and (ii) the user modifying the content of the first file stored on the first client device.

Petition for *Inter Partes* Review of
U.S. Patent No. 10,754,823

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 13,832 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. 10,754,823

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. 10,754,823**, 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
100 Congress Avenue, Suite 2000
Austin, TX 78701

DATED: January 4, 2023

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