UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

DROPBOX, INC.
Petitioner

v.

MOTION OFFENSE, LLC
Patent Owner

IPR2024-00286
U.S. Patent No. 11,611,520

DECLARATION OF TODD MOWRY, PH.D.
U.S. PATENT NO. 11,611,520
CHALLENGING CLAIMS 1–8 and 17–21

TABLE OF CONTENTS

Page

I. INTRODUCTION 1

II. QUALIFICATIONS AND PROFESSIONAL EXPERIENCE 1

III. MATERIALS CONSIDERED 5

IV. LEGAL PRINCIPLES 7

A. Claim Construction 7
B. Anticipation 8
C. Obviousness 8

V. LEVEL OF ORDINARY SKILL 11

VI. TECHNICAL BACKGROUND 12

A. Internet-Based Data Storage Systems 12
B. Accessing Internet-Based Data Storage Systems 13

VII. OVERVIEW OF THE ’520 PATENT 16

A. Alleged Invention 16
B. Prosecution History 17

VIII. OVERVIEW OF THE PRIOR ART REFERENCES 19

A. Houston 19
B. Garcia 23
C. Manzano 28
D. Wu 31

IX. SPECIFIC GROUNDS FOR UNPATENTABILITY 32

A. Ground 1: Claims 1–8 and 17–21 are Obvious over Houston and Garcia 32
1. Independent Claim 1 32
2. Dependent Claim 2 62
3. Dependent Claim 3 64
4. Dependent Claim 4 67

5. Dependent Claim 5 70
6. Dependent Claim 6 72
7. Independent Claim 7 73
8. Dependent Claim 8 79
9. Independent Claim 17 79
10. Dependent Claim 18 86
11. Dependent Claim 19 87
12. Dependent Claim 20 89
13. Dependent Claim 21 89

B. Ground 2: Claims 1–8 and 17–21 are Obvious over Houston, Garcia, and Manzano 90
C. Grounds 3 and 4: Claims 17–21 are Obvious over Houston, Garcia, and Wu or over Houston, Garcia, Manzano, and Wu 99

X. CONCLUSION 103

XI. AVAILABILITY FOR CROSS EXAMINATION 103

XII. RIGHT TO SUPPLEMENT 103

XIII. JURAT 104

I, Todd Mowry, declare as follows:

I. INTRODUCTION

1. My name is Todd Mowry.

2. I have been retained as an expert in this proceeding by counsel for Dropbox, Inc. ("Dropbox") to provide my expert conclusions regarding the validity of claims 1–8 and 17–21 of U.S. Patent No. 11,611,520 (the "'520 patent"). For the reasons set forth below, it is my conclusion that claims 1–8 and 17–21 of the '520 patent are invalid.

II. QUALIFICATIONS AND PROFESSIONAL EXPERIENCE

3. My qualifications are stated more fully in my curriculum vitae, which is attached as Exhibit A. Below is a summary of my education, work experience, and other qualifications.

4. I am a Professor in the Department of Computer Science at Carnegie Mellon University. I also have a courtesy appointment in the Department of Electrical and Computer Engineering. I have served on the faculty of Carnegie Mellon University for 26 years starting in 1997 through the present (2023).

5. I also served on the faculty of the University of Toronto for four years between 1993 and 1997, in the Department of Electrical and Computer Engineering and a courtesy appointment in the Department of Computer Science. Prior to that

appointment, I served as a Graduate Research Assistant in the Department of Electrical Engineering at Stanford University for four years between 1989 and 1993.

6. As a faculty member, I have taught and continue to teach courses and directed research in computer systems and software, operating systems, distributed and network systems, parallel processing, and compiler optimization.

7. I received a B.S. degree in Electrical Engineering with Highest Distinction from the University of Virginia in May 1988. I received an M.S. in Electrical Engineering from Stanford University in June 1989, and a Ph.D. in Electrical Engineering from Stanford University in March 1994.

8. I have worked in the computer industry in various capacities. I was a part-time Computer Architect and then Computer Architecture Consultant at Silicon Graphics, Inc., in Mountain View, California (formerly MIPS Computer Systems in Sunnyvale, California) from 1989 to 1993 and 1993 to 1996, respectively. I was a Visiting Scientist at IBM in Toronto from 1996 to 2004. During that same time period (1996 to 2004), I was also a Member of the Technical Advisory Board of SandCraft, Inc. in Santa Clara, California. I was the Director of the Intel Research Pittsburgh Lab at Intel Corporation in Pittsburgh, Pennsylvania from 2004 to 2007.

9. I have authored 20 journal articles and 69 conference papers. I am also an inventor on 5 patents.

10. I have published a number of papers in the top research conferences in the fields of operating systems and data storage (including a paper that won the Best Paper Award at the USENIX 2nd Symposium on Operating Systems Design and Implementation (OSDI '96)). *See, e.g.*, Olatunji Ruwase, Michael A. Kozuch, Phillip B. Gibbons, and Todd C. Mowry, *Guardrail: High Fidelity Correctness Checking of Device Drivers for Safeguarding I/O Operations*, in Proceedings of the Nineteenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), (2014); Andrew Pavlo, Gustavo Angulo, Joy Arulraj, Haibin Lin, Jiexi Lin, Lin Ma, Prashanth Menon, Todd Mowry, Matthew Perron, Ian Quah, Siddharth Santurkar, Anthony Tomasic, Skye Toor, Dana Van Aken, ZiqiWang, YingjunWu, Ran Xian, and Tieying Zhang, *Self-Driving Database Management Systems*, in Proceedings of the 2017 Conference on Innovative Data Systems Research (CIDR), (2017); and Amit Manjhi, Charles Garrod, Bruce M. Maggs, Todd C. Mowry, Anthony Tomasic, *Holistic Query Transformations for Dynamic Web Applications*, in 2009 IEEE 25th International Conference on Data Engineering (ICDE) (2009).

11. I have also been an active member of the Parallel Data Lab at Carnegie Mellon University since 1997, which is often described as "academia's premiere storage systems research center."

12. I am the recipient of several honors and awards: the Arthur Samuel Thesis Award (awarded by the Stanford Computer Science department to the top two Ph.D. theses in a given year), several IBM Faculty Development Awards (1996, 1997, 1998, 2000, 2001, 2002, and 2003), several Best Paper Awards (the Second Symposium on Operating Systems Design and Implementation in 1996; the 20th International Conference on Data Engineering (ICDE) in 2004), the Alfred P. Sloan Research Fellowship (awarded to researchers in recognition of distinguished performance and a unique potential to make substantial contributions to their field), the Most Thought-Provoking Idea Award in 2004 (awarded by the Architectural Support for Programming Languages and Operating Systems (ASPLOS)), the TR100 Award in 1999 (awarded by MIT's Technology Review magazine to the top 100 most promising young innovators in science and technology), and I became an ACM Fellow in 2016.

13. I am a member of the Institute of Electrical and Electronics Engineers (IEEE) and the Association of Computing Machinery (ACM). I was the Editor-in-Chief of ACM Transactions on Computer Systems (the premier journal for computer systems research) from 2013-2018. I was an Associate Editor for the journal prior to that, since 2001. I was the Program Chair of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS) in 2010. I was the Co-Program Chair of the International Conference on

Parallel Architectures and Compilation Techniques (PACT) in 2001. I have been on the programming committee in various years for ASPLOS, the International Symposium on Computer Architecture (ISCA), the International Symposium on Microarchitectures, and the Workshop on Architectural and System Support for Improving Format.

14. Overall, I have over 29 years of experience in the field of computer science and, specifically, computer architecture, operating systems, and document storage and retrieval.

15. In light of the foregoing, I consider myself to be an expert in the field of computer science, and believe that I am qualified to provide an opinion as to what a person of ordinary skill in the art would have understood, known, or concluded regarding the subject matter of the ’520 patent at the time of its alleged invention.

16. I am being compensated for my time at my ordinary hourly rate of $650. My compensation is not dependent on the outcome of these proceedings or the content of my opinions. To the best of my knowledge, I have no financial interest in Dropbox. I have no financial interest in the ’520 patent.

III. MATERIALS CONSIDERED

17. In preparing this declaration, I have reviewed the claims, specification, and file history of the ’520 patent.

18. I have reviewed and understand the references and exhibits cited in this Declaration, including the following:

Exhibit	Description
1001	U.S. Patent No. 11,611,520
1003	U.S. Patent No. 8,825,597 (“Houston”)
1004	U.S. Patent No. 9,633,125 (“Garcia”)
1005	U.S. Patent Application Publication No. 2010/0005138 (“Manzano”)
1006	U.S. Patent Application Publication No. 2007/0011246 (“Wu”)
1008	U.S. Patent Application Publication No. 2012/0278404 (“Meisels”)
1009	U.S. Patent Application Publication No. 2013/0067303 (“Kantor”)
1011	“SugarSync now allows you to work directly in Windows Explorer,” D. Mihovilovic, May 6, 2009
1012	Internet Archive Declaration of Nathaniel E Frank-White
1014	’520 Patent File History
1015	U.S. Patent No. 6,351,776 (“O’Brien”)
1016	U.S. Patent Application Publication No. 2008/0320397 (“Do”)
1017	“Supporting File Sharing Through Improved Awareness,” T. Whalen, August 2008

IV. LEGAL PRINCIPLES

19. I am not an attorney. For purposes of this declaration, I have been informed by Dropbox's counsel about certain aspects of the law that are relevant to my analysis and opinions, as stated in the following paragraphs. I have applied these legal principles in arriving at my conclusions expressed in this declaration.

A. Claim Construction

20. I understand that "claim construction" is the process of determining a patent claim's meaning. I have been informed that claim construction is a matter of law and that in this proceeding, claim construction will be determined by the Patent Trial and Appeal Board ("the Board"). I understand that, during IPR, claims are construed according to the "*Phillips* standard." I understand that the Board need only construe the claims when necessary to resolve the underlying controversy. I also understand that, given the close correlation between the asserted prior art and the challenged claims of the '520 patent, the Board need not construe any terms of the challenged claims to resolve the underlying controversy, as any reasonable interpretation of those terms consistent with their plain meaning (as would have been understood by a POSITA at the time of the invention, having taken into consideration the language of the claims, the specification, and the prosecution history of record) reads on the prior art.

B. Anticipation

21. I have been informed and understand that a claim is invalid if it was anticipated or not novel. I also have been informed and understand that a claim is anticipated if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.

C. Obviousness

22. I have been informed and understand that a patent claim is invalid if it would have been obvious to a person of ordinary skill in the art ("POSITA") at the time of the alleged invention. This means that, even if all of the requirements of a claim are not found in a single prior art reference, the claim is not patentable if the differences between the subject matter in the prior art and the subject matter in the claim would have been obvious to a POSITA at the time of the alleged invention.

23. I have been informed and understand that a determination of whether a claim was obvious should be based upon several factors, including, among others: (1) the level of ordinary skill in the art at the time of the alleged invention; (2) the scope and content of the prior art; (3) the differences, if any, that existed between the claimed invention and the prior art; and (4) any objective considerations of nonobviousness.

24. I have been informed and understand that when determining any differences between the invention covered by the patent claims and the prior art, one

should not look at the individual differences between the patent claims and the prior art in isolation, but rather consider the claimed invention as a whole and determine whether or not it would have been obvious in light of all of the prior art.

25. I have been informed and understand that a single reference can render a patent claim obvious if any differences between that reference and the claims would have been obvious to a POSITA. Alternatively, the teachings of two or more references may be combined in the same way as disclosed in the claims. I have been informed and understand that when deciding whether to combine the various items described in the prior art, the relevant question is whether the prior art combination would have been obvious to a person with ordinary skill in the art at the time of the invention. I have been further informed that it can be important to identify a teaching, suggestion, or motivation, in either the prior art or the knowledge of persons skilled in the art, that would have prompted a person of ordinary skill in the relevant field to combine the elements in the prior art in the way the claimed invention does.

26. In determining whether a combination based on either a single reference or multiple references would have been obvious, it is appropriate to consider, among other factors:

- whether the teachings of the prior art references disclose known concepts combined in familiar ways, and when combined, would yield predictable results;
- whether a POSITA would implement a predictable variation, and would see the benefit of doing so;
- whether the claimed elements represent one of a limited number of known design choices, and would have a reasonable expectation of success by a POSITA;
- whether a POSITA would have recognized a reason to combine known elements in the manner described in the claim;
- whether there is some teaching or suggestion in the prior art to make the modification or combination of elements claimed in the patent; and
- whether the innovation applies a known technique that had been used to improve a similar device or method in a similar way.

27. I have been informed and understand that a POSITA has ordinary creativity, and is not an automaton. I have been informed and understand that in considering obviousness, it is important not to determine obviousness using the benefit of hindsight derived from the patent being considered. I have been informed and understand that a patent composed of several elements is not proved obvious

merely by demonstrating that each of its elements was independently known in the prior art.

28. I have been informed and understand that to show a motivation to combine, one is not required to show how the references would be physically combined to render a claim obvious. The analysis is not focused on whether the features of one reference can be physically incorporated into the structure of another reference, but rather whether a POSITA would have been motivated to combine the teachings of the references to achieve the claimed invention. For example, a POSITA would understand how predictable variations of known concepts could be combined and would recognize that were where multiple tools were available in a toolbox, any could be chosen.

V. LEVEL OF ORDINARY SKILL

29. I have considered the '520 patent from the perspective of a POSITA at the time of the alleged invention. A POSITA for the '520 patent would have had at least the equivalent of a Bachelor's degree in electrical engineering, computer science, or a related field and two or more years of experience in a related field such as networked computer systems using Internet protocols. Less work experience may be compensated by a higher level of education, such as a Master's Degree, and vice versa. At the time of the claimed invention, I would have qualified as a POSITA under this standard.

VI. TECHNICAL BACKGROUND

A. Internet-Based Data Storage Systems

30. As early as the year 2000, more than a decade before the earliest claimed priority date of the '520 patent, there was a burst in Internet-based (or "cloud") data storage systems. *See, e.g.*, U.S. Patent No. 6,351,776 ("O'Brien"; Exhibit 1015), 1:10–3:4; U.S. Patent Application Publication No. 2008/0320397 ("Do"; Exhibit 1016), ¶¶ 1–4; U.S. Patent Application Publication No. 2013/0067303 ("Kantor"; Exhibit 1009), ¶ 1. As O'Brien explained at the time, because personal computers ("PCs") were not easily transportable and laptops had fewer resources than desktops, there was a need for "means by which files and other data may be available worldwide through the Internet and not tied to a local computer." Exhibit 1015, 2:42–3:1.

31. With the increased availability of broadband access and widespread use of the Internet, computer users increasingly started to store "their electronic documents or files on remote computer servers that are geographically remote from the user." Exhibit 1016, ¶ 1; *see also* Exhibit 1009, ¶ 1 ("The availability and popularity of web-based content and services is ever increasing. For example, users are increasingly using online storage services that provide storage 'in the cloud' for content items (e.g., documents, application files, photos, mobile uploads, audio/video files, etc.)."). Do noted that "[s]toring files on a remote server provides

users with several benefits" such as backup copies of files, data recovery in the event of a computer crash or a disaster, and access to files "from any computer with an Internet connection." Exhibit 1016, ¶ 2. Notably, users storing files on a remote server could share their files with others by providing them with access to the server. *Id.*, ¶ 4; *see also* Exhibit 1015, 4:1–5 ("The user can . . . give permission to share files on the Shared Internet Storage Resource with others."); Exhibit 1009, ¶ 1 ("Due in part to this convenience, users may also seek ways in which to share content items from their online storage with others.").

B. Accessing Internet-Based Data Storage Systems

32. It was well known that a user could use not only a web-based application in a web browser but also a client-based application to access files stored in a remote server. For example, O'Brien describes, and illustrates in Figure 13 (reproduced below), an online storage system ("X:Drive") that is accessible from a "**web-based application 1350**," a "**'My Computer' window 1310**," and a "**client-based application window 1320**." Exhibit 1015, 20:21–52.

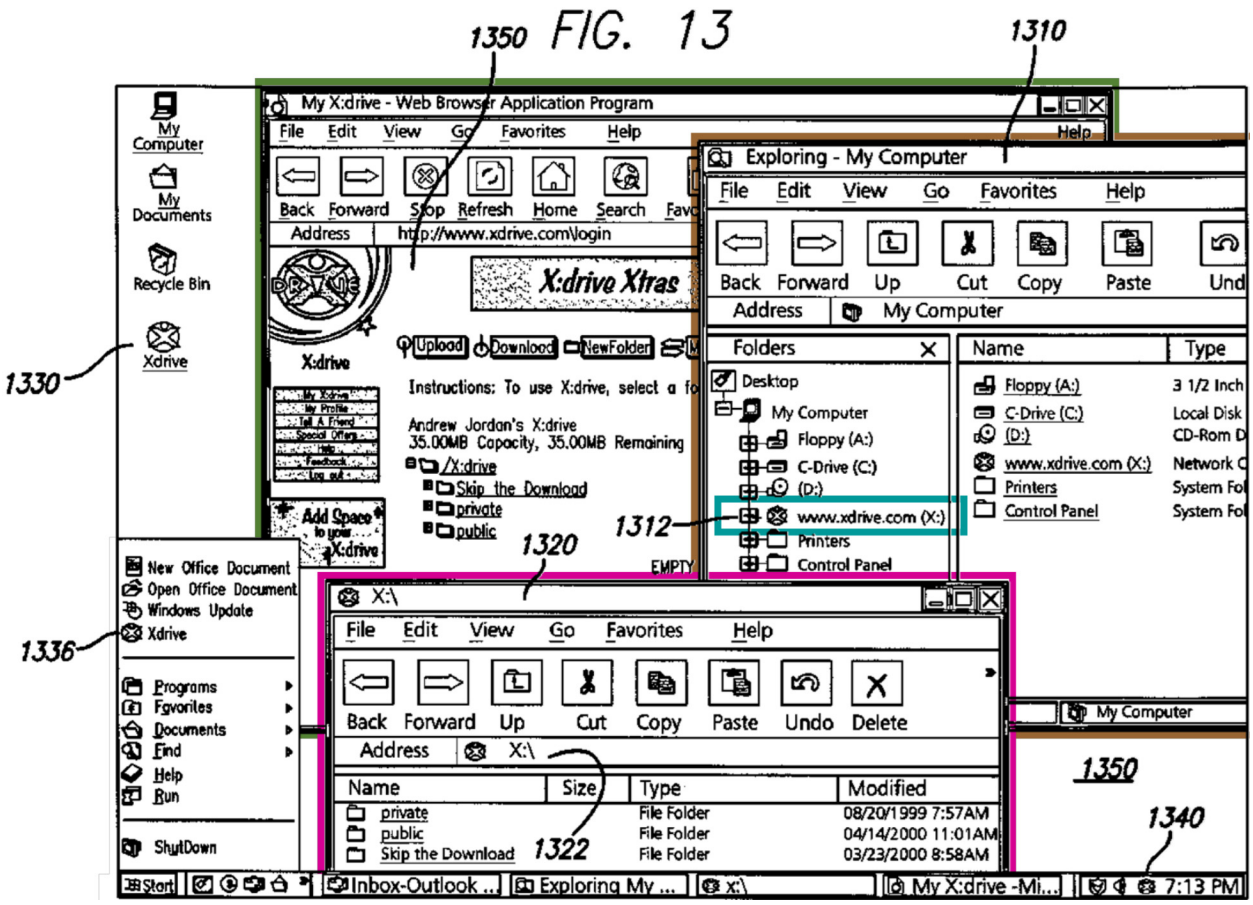

Exhibit 1015, Figure 13 (annotated)

33. The **"My Computer" window 1310** is part of the operating system of the user's computer (*e.g.*, Windows Explorer or File Explorer from Microsoft Windows, Finder from Mac, etc.) and includes a "**drive at logical letter X: 1312**," which is provided by a client application/program 102 from the online storage system provider and residing on the user computer's operating system. *Id.*, 20:25–31, 7:5–8; *see also* Exhibit 1003, Figure 10 ("My Dropbox" folder 1002); Exhibit 1004, Figure 10 ("SugarSync Drive" virtual drive 1010), 7:22–24 ("a user interface 2170 (e.g., MAC FINDER or WINDOWS EXPLORER) which displays the physical

drive user interface 2180 and the virtual drive user interface 2190."). O'Brien's **drive 1312** appears among other local drives and "is transparent to the user and functions as any other drive present on the system." Exhibit 1015, 20:29–31.

34. It was also well known that, because local storage on computers was limited (mainly for mobile devices such as laptops), a file stored on a remote server and displayed in a client-based file explorer interface did not have to be downloaded from the server until the user clicks to open the file. *See, e.g.*, Exhibit 1004, 5:35–39 ("when a user double clicks on a file or item in a remote folder in the virtual drive, the server downloads its copy of the file/item to the local synchronization software client (i.e., the client currently being used by the user)."); Exhibit 1005, ¶ 50 ("If the corresponding file (i.e., file to be shared by source client 310) is not local to user device 280, the icon 600 may be ghosted (i.e., semi-transparent) or otherwise be visually configured to indicate the non-local status of the file."), ¶ 51 ("a user may right-click or otherwise indicate the icon 600 . . . to request receipt or download of the associated file, at which time the icon 600 will become unghosted or will otherwise be visually configured to indicate the pending-local or local status of the file.").

VII. OVERVIEW OF THE '520 PATENT

A. Alleged Invention

35. The '520 patent generally relates to "[m]ethods, systems, and computer program products for processing a data object identification request in a communication" between a "first node" and a "second node." Exhibit 1001, Title, Abstract. At best, the specification of the '520 patent purports to describe techniques for requesting a file, but is devoid of how to conduct folder sharing, as recited by claims 1–8 and 17–21.

36. According to independent claims 1, 7, and 17, "at least one folder"[1] is shared between "a first node" and "a second node" via "at least one network." *See* Exhibit 1001, claims 1, 7, and 17. At the first node, a series of interfaces and user interface elements are displayed to allow the at least one folder to be selected and an email address of a user of the second node to be specified. *See id.* Once the folder sharing is initiated, an email message is generated and sent to the second node. *See id.* The email message includes a reference to the at least one folder and does not include "at least one file" in the at least one folder as an attachment. *See id.* A "representation of the at least one folder" is also created among other folders in a

[1] Claim 7 recites "at least one data object." As I explain later below, a data object includes a folder.

"file explorer interface" at the second node. *See id.* In some embodiments, the second node does not store the at least one file when the representation of the at least one folder is created in the file explorer interface. *See id.*, claims 1 and 7. Upon detection of an indication to open the at least one file, the at least one file is retrieved via the at least one network and displayed at the second node. *See id.* In other embodiments, another representation of the at least one folder is created and displayed in a "web page." *See id.*, claim 17.

37. As I will demonstrate in the grounds below, sharing a folder and accessing a file in a shared folder in such a fashion were well-known in the art before the '520 patent's earliest possible priority date.

B. Prosecution History

38. The '520 patent issued from U.S. Patent Application No. 17/187,621, which was filed on February 26, 2021. The '520 patent is a continuation of U.S. Patent No. 11,044,215, which is a continuation of U.S. Patent No. 10,587,548, which is a continuation of U.S. Patent No. 10,021,052, which is a continuation-in-part (CIP) of application No. 14/274,623, which is a CIP of application Nos. 13/626,635 and 13/624,906.

39. The examiner issued a sole Office Action on October 15, 2021, rejecting all originally-filed claims 1–20 under (1) 35 U.S.C § 101 statutory double patenting as claiming the same invention as the '215 patent, (2) the ground of

nonstatutory double patenting as being unpatentable over the claims of the ’052 patent, and (3) the ground of nonstatutory double patenting as being unpatentable over the claims of the ’548 patent. *See* Exhibit 1014 at 194–220. In response, on March 14, 2022, the applicant filed a terminal disclaimer over the ’052 and ’548 patents and made minor amendments to the claims, without any substantive argument. *See id.* at 321–322, 326–339.

40. On April 4, 2022, the examiner issued a Notice of Allowance (NoA). *See* Exhibit 1014 at 407–415. The applicant filed a Request for Continued Examination (RCE) on July 1, 2022 along with a minor amendment to original claim 19. *See id.* at 466–481. On July 11, 2022, the applicant filed a Supplemental Amendment, cancelling claims 1–20 and adding new claims 21–44. *See id.* at 492–511.

41. On August 9, 2022, the examiner issued a second NoA, essentially stating that the “closest prior art” references (Plestid, Cook, and Daniels, which are not relied on the grounds below) do not teach most of the limitations of the independent claims. *See* Exhibit 1014 at 539–553. The applicant filed yet another RCE on August 16, 2022, with a Track One Request and an Information Disclosure Statement (IDS). *See id.* at 591–617.

42. On September 19, 2022, the examiner issued a third NoA. *See* Exhibit 1014 at 706–713. The applicant paid the issue fee on September 20, 2022, but

subsequently filed an RCE on November 18, 2022 with an IDS (and another Track One Request, which was denied), and petitioned to withdraw the application from issue on November 28, 2022. *See id.* at 833, 838–839, 846, 857, 862.

43. On February 13, 2023, the examiner issued a fourth NoA. *See* Exhibit 1014 at 868–872. The applicant filed an issue fee transmittal on February 14, 2023. *See id.* at 886. The application issued as the '520 patent on March 21, 2023. At no time during prosecution did the examiner provide evaluation of the claims in view of any prior art.

VIII. OVERVIEW OF THE PRIOR ART REFERENCES

A. Houston

44. U.S. Patent No. 8,825,597 ("Houston"; Exhibit 1003) "relates generally to sharing of data over a network." Exhibit 1003, 1:14–15. Houston notes that it was conventional for people to collaborate by exchanging files as "e-mail attachments" but that attaching files is "cumbersome." *Id.*, 1:36–45. Accordingly, Houston discloses a system that synchronizes folders and their contents shared among multiple clients (*e.g.*, clients 108a and 108b) with a host system 110 over a network. *See id.*, Abstract, 1:15–17, 1:53–56, Figure 1 (reproduced below).

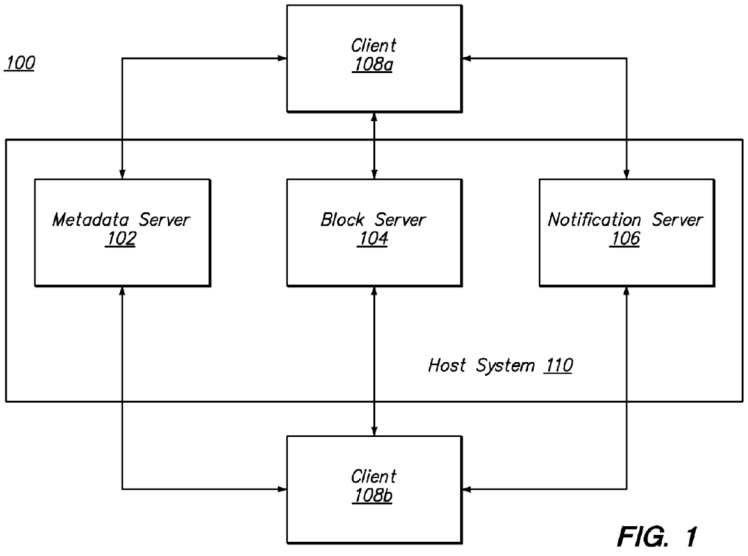

FIG. 1

Exhibit 1003, Figure 1

45. In particular, Houston's host system 110 provides, via a web interface, a **user interface window 200** for a user to "create and share a new folder" by specifying the name of the folder in a **text box**. Exhibit 1003, Figure 2 (reproduced below), 3:34–38.

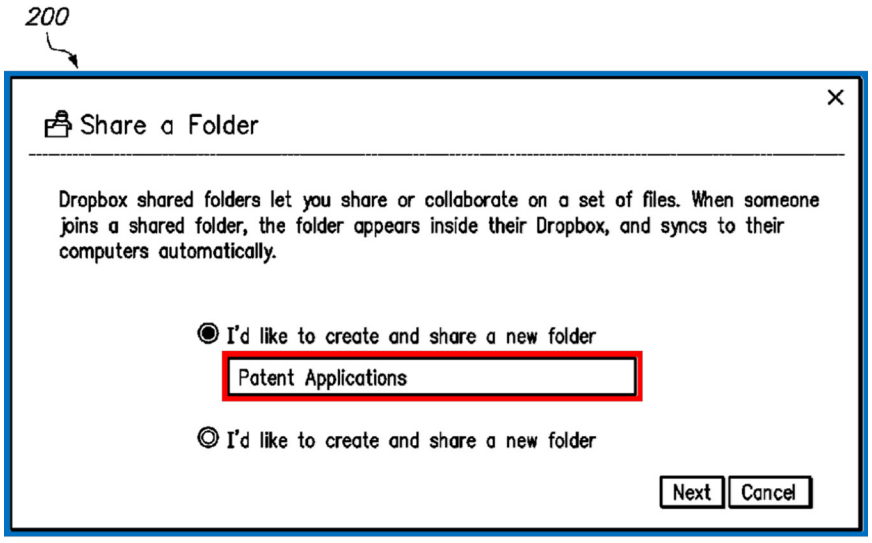

FIG. 2

Exhibit 1003, Figure 2 (annotated)

46. Alternatively, the user can choose to share an existing folder among existing folders in a **file structure** presented in a **user interface window 300**. *Id.*, Figure 3 (reproduced below), 3:38–40.

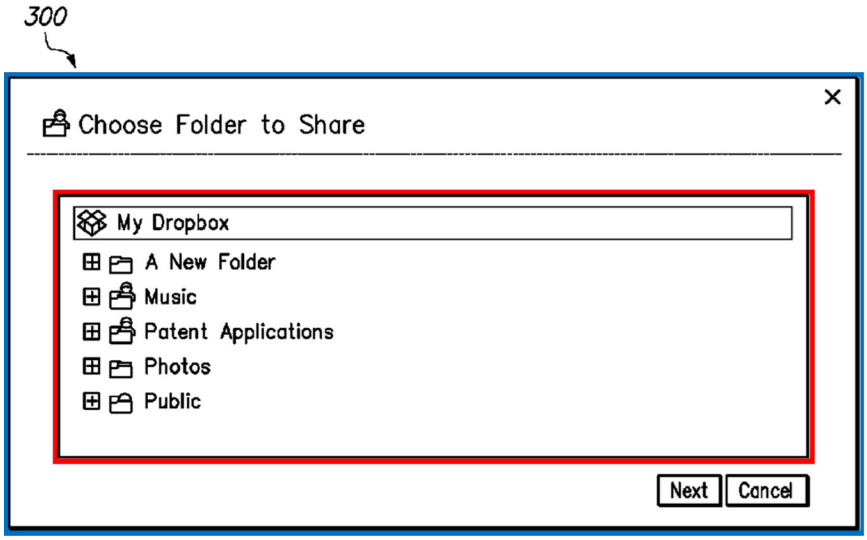

FIG. 3

Exhibit 1003, Figure 3 (annotated)

47. Once the user has chosen or created the folder to be shared, the user is presented with a **user interface window 400** that allows the user to invite others to the shared folder by entering their **email addresses** in a **text box** and clicking a **"Share folder" button**. *See* Exhibit 1003, Figure 4 (reproduced below), 3:40–43.

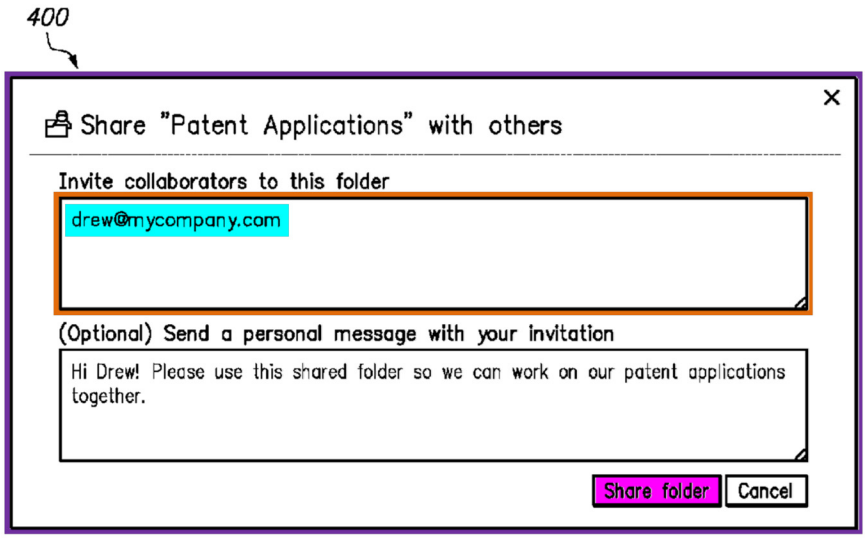

FIG. 4

Exhibit 1003, Figure 4 (annotated)

48. Houston further discloses an **interface**, local to a client device, "for interacting with shared folders." *Id.*, Figure 10 (reproduced below), 2:48–50. Houston describes that the **"My Dropbox" folder 1002** is "a folder that is synchronized with host system 110" and "is displayed next to other conventional folders such as 'My Meetings', 'My Received Files,' and others." *Id.*, 7:21–31. The **"Patent Applications" folder**, with a **"people" icon**, is an example of a shared folder displayed among other folders in the **interface**. *See, e.g.*, Exhibit 1011 ("shared folders are shown with a blue 'people' icon").

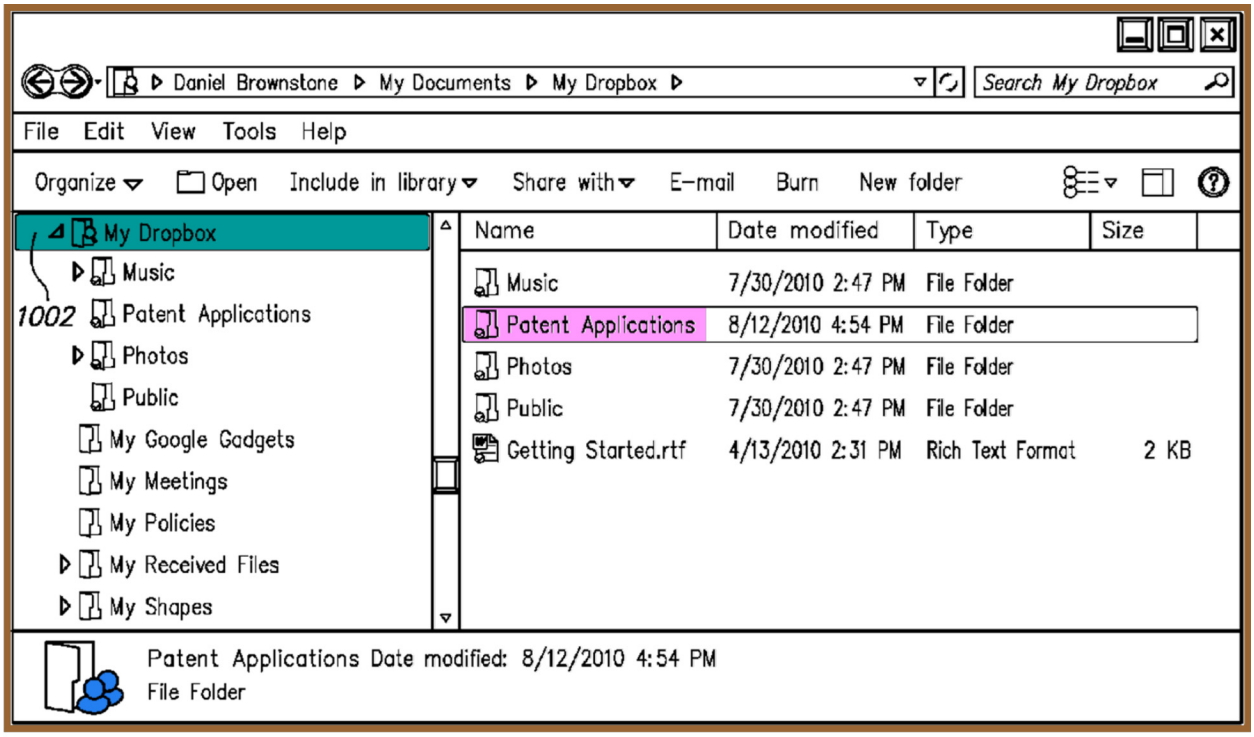

Exhibit 1003, Figure 10 (annotated)

49. In other embodiments, Houston discloses that "a web interface allows access to shared folders from any computer with network access." Exhibit 1003, 2:10–12.

B. Garcia

50. U.S. Patent No. 9,633,125 to Garcia et al. ("Garcia"; Exhibit 1004) relates to "a system and method for enabling a user to synchronize, manage, and share folders across a plurality of client devices and a synchronization server." Exhibit 1004, 1:12–15; *see also id.*, Abstract, 1:39–42. For example, Garcia discloses a synchronization system 200 with "a plurality of client devices" (*e.g.*, home computer 230, work computer 240, mobile device 250) in communication with

a synchronization server 220 via a network 210. *Id.*, Figure 2 (reproduced below), 8:8–26 ("client devices 230, 240, 250 may all belong to the same user or may belong to different users who have shared access to objects with one another"), 7:54–55 ("an object may be a folder or a file").

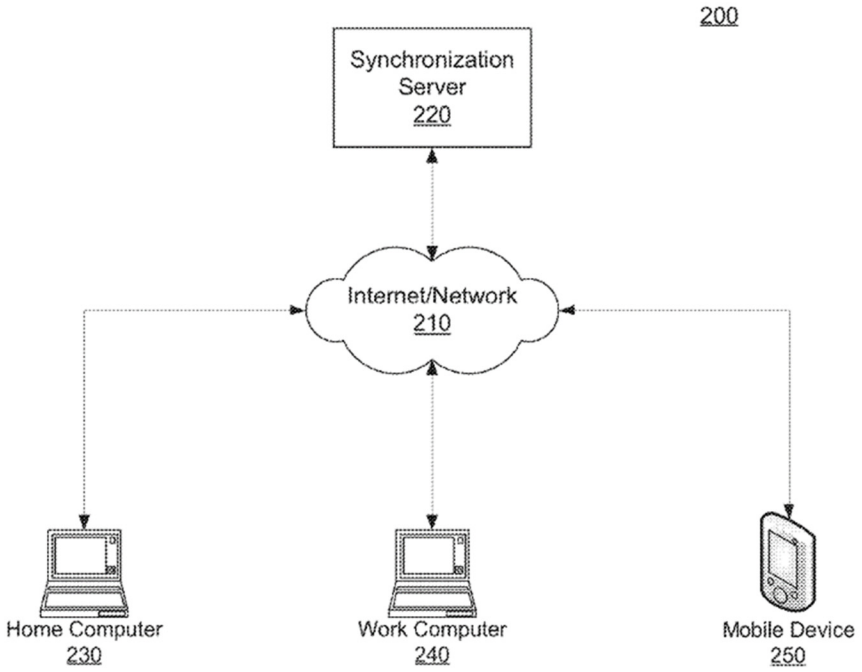

Exhibit 1004, Figure 2

51. Correspondingly, Garcia discloses a synchronization system 2100 that includes a synchronization server 2120 in communication with a client device 2130 via an internet/network 2110. *See id.*, Figure 21 (reproduced below), 6:36–67. Garcia describes that "all synchronized objects between a user's client devices and between users are stored on server 2120 and are accessible from any client device having access to the server 2120." *Id.*, 6:45–48; *see also id.*, 2:37–41 ("A user may be able to access, via [a] user interface, a virtual drive on the local device, where,

via the virtual drive, a user is able to access items (e.g., files) in any object (e.g., folders), including objects located only on a remote device.").

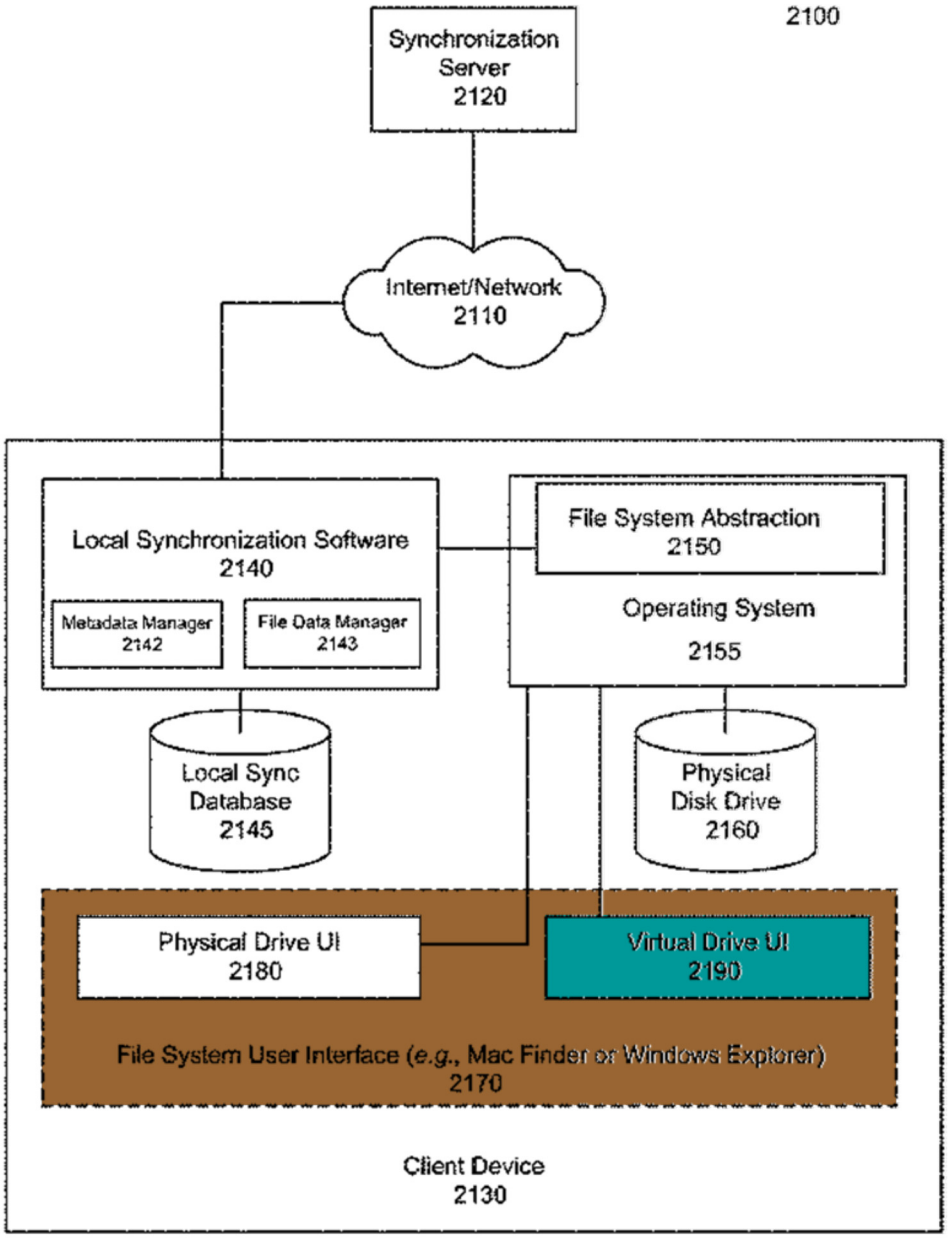

Exhibit 1004, Figure 21 (annotated)

52. Garcia explains that "[w]ithin client device 2130 is local synchronization client software 2140." *Id.*, 7:1–2. The local synchronization client

software 2140 "functions to synchronize client device 2130 with the synchronization server 2120." *Id.*, 7:2–3. Moreover, "[t]he local synchronization software 2140 is operatively coupled to a file system abstraction module 2150, within operating system 2155" of the client device 2130. *Id.*, 7:22–24. The operating system 2155 in turn is "connected to a **user interface 2170** (e.g., MAC FINDER or WINDOWS EXPLORER) which displays the physical drive user interface 2180 and the **virtual drive user interface 2190.**" *Id.*, 7:25–29.

53. Consistent with its description of Figure 21, Garcia illustrates a **virtual drive 1010** displayed in an **interface** in Figure 10 (reproduced below). Garcia explains that "[t]he **virtual drive** appears as a separate disk drive on the user's local client device" in the **interface**. *Id.*, 5:27–30.

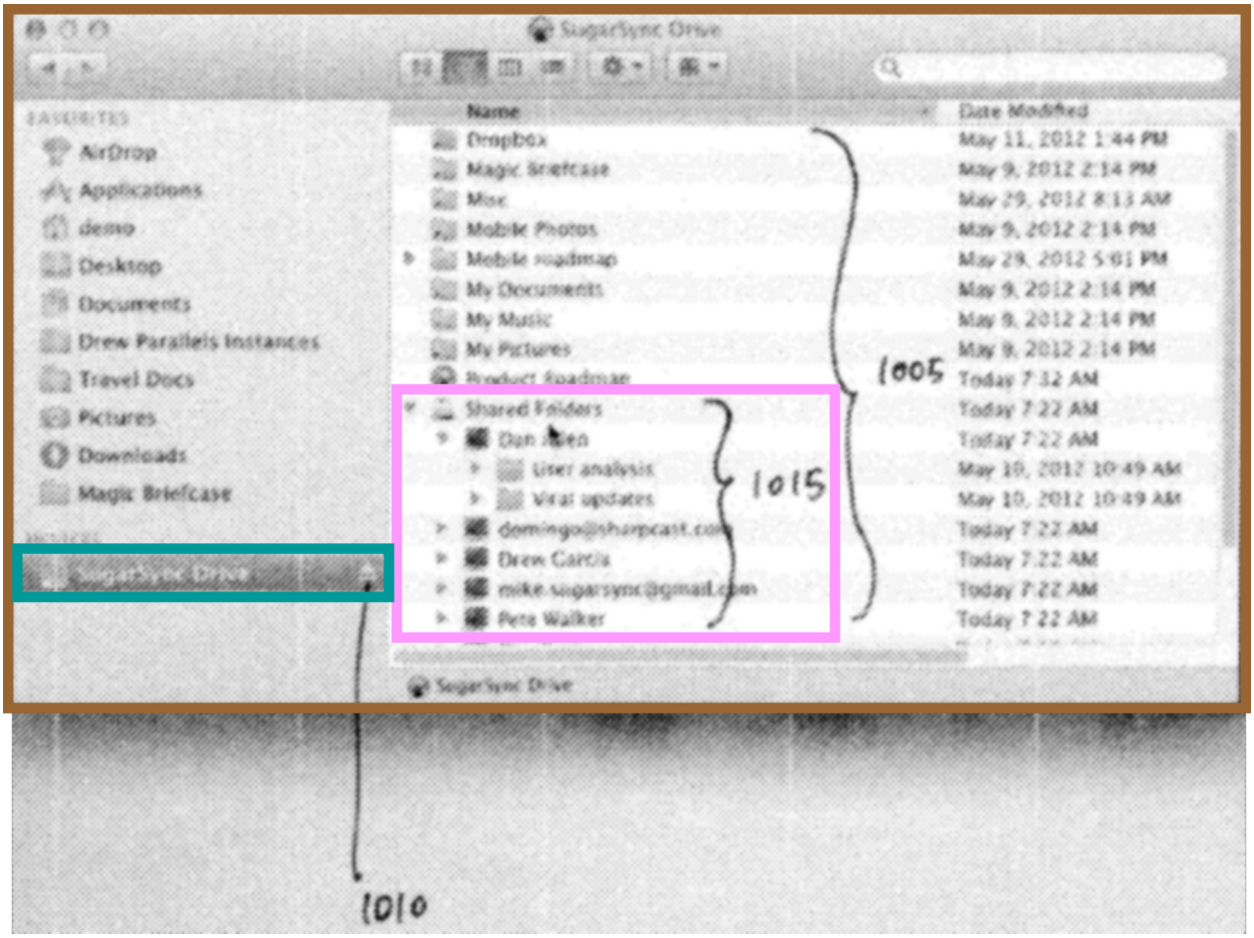

Exhibit 1004, Figure 10 (excerpted and annotated)

54. Garcia discloses that "any file in folders 1005 in the synchronization system may be accessed through [the] **virtual drive 1010**" and that "[n]ot only are all of the user's folders that have been synchronized to or stored on server 220 accessible through the virtual drive, but also all of the **folders 1015 shared with the user by other users**." *Id.*, 10:59–63, 11:7–10. Garcia goes on to explain that, "when a user double clicks on a file or item in a remote folder in the virtual drive, the server downloads its copy of the file/item to the local synchronization software client (i.e., the client currently being used by the user)." *Id.*, 5:35–39.

55. Garcia also teaches that a client can share a folder with another client by generating and sending an email with a link to and identifying the shared folder, instead of an attachment. *See* Exhibit 1004, 10:32–34 ("The Web link tab 335 includes a list of folders or files that have been shared as a link with other users."), 10:38–40 ("[A]n icon 915 . . . places the link onto the clipboard to be copied into an email, text, or other transmittal."), 11:39–43, claim 3, Figures 9, 11–12.

C. Manzano

56. U.S. Patent Application Publication No. 2010/0005138 to Manzano (Exhibit 1005) relates to "sharing of electronic files among electronic devices." Exhibit 1005, ¶ 2. Manzano discloses a computer network system 200 comprising a server 230 linked to user devices 210, 280 via a network 260. *Id.*, Figure 2 (reproduced below), ¶ 29. "User devices 210, 280 include or are otherwise coupled to a computer screen or display 250, 290, respectively." *Id.*, ¶ 30.

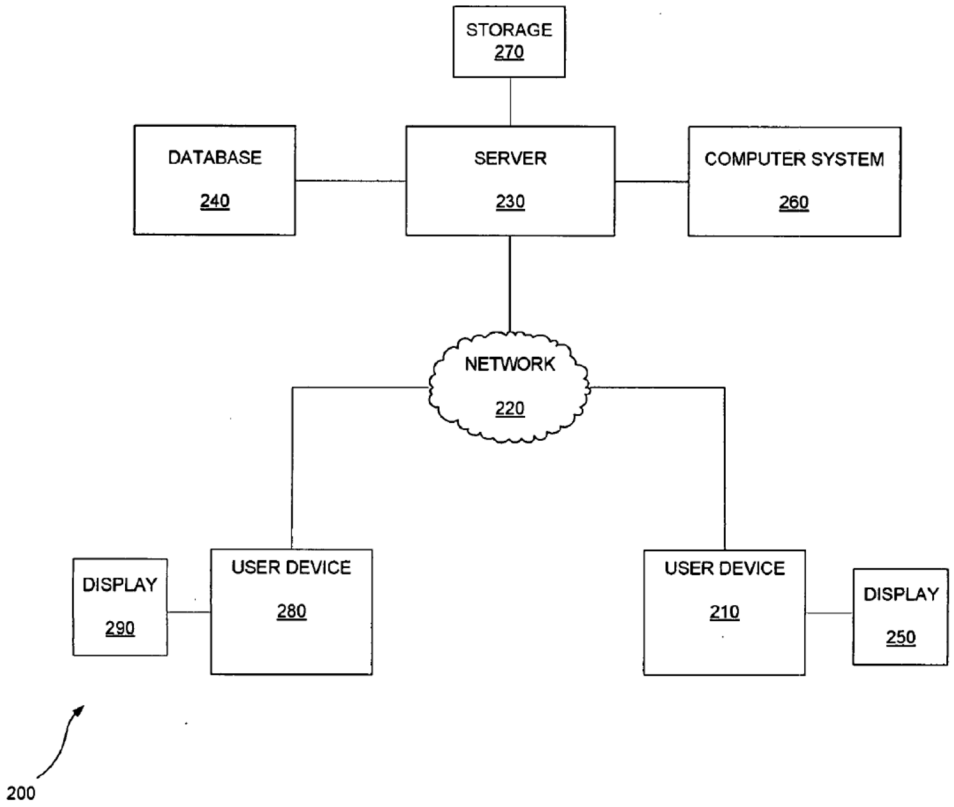

Exhibit 1005, Figure 2 (annotated)

57. Manzano discloses that "a file-transfer application" may be transferred from the server 230 to the user devices 210, 280 and that "[o]nce installed on the user devices 210, 280, the file-transfer application provides file-transfer clients 310, 320 executable by the user devices 210, 280, respectively." *Id.*, ¶ 32; *see also id.*, Figure 3 (reproduced below). Manzano discloses that, "[a]lso included in the file-transfer application are user interfaces 350, 360 that are displayable on the displays 250, 290, respectively" and that "the interfaces 350, 360 correspond to respective electronic folders or windows into which files may be placed." *Id.*

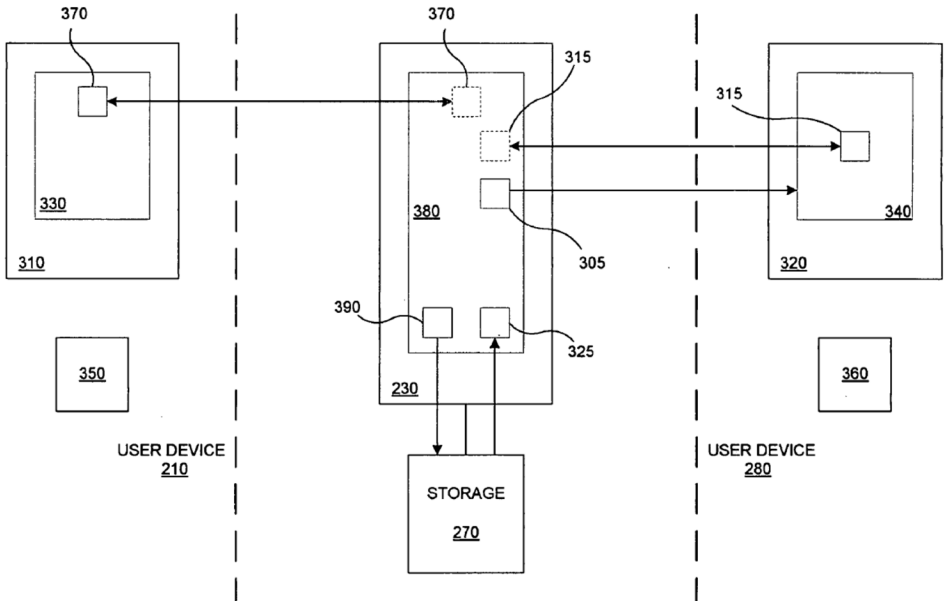

Exhibit 1005, Figure 3 (annotated)

58. Manzano discloses that "file icons, such as the icon 600, in a workspace displayed on display device 290 can visually indicate whether a corresponding file (such as a file, the transfer of which may be as is discussed herein with reference to Figure 3) has been downloaded to or is otherwise stored on (i.e., is local to) . . . the user device 280." *Id.*, ¶ 50; *see also id.*, ¶ 52 ("'workspace' corresponds to a respective electronic folder"). Manzano explains that, if a shared file "is not local to user device 280, the icon 600 may be ghosted (i.e., semi-transparent) or otherwise be visually configured to indicate the non-local status of the file." *Id.*, ¶ 50.

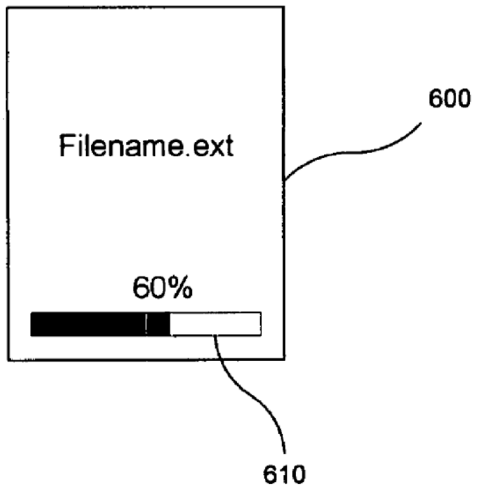

Exhibit 1005, Figure 6

59. Thereafter, according to Manzano, at the request of a user of the user device 280, the file may be downloaded, after which the icon 600 will become "unghosted." *Id.*, ¶ 51 ("[A] user may right-click or otherwise indicate the icon 600, and from a resultant pop-up contextual menu (not shown), for example, that appears on the display 290, make a selection from the menu to request receipt or download of the associated file, at which time the icon 600 will become unghosted or will otherwise be visually configured to indicate the pending-local or local status of the file.").

D. Wu

60. U.S. Patent Application Publication No. 2007/0011246 to Wu (Exhibit 1006) discloses "[a] system and method of producing E-mail [by using templates to] reduce the time consumed for producing E-mail." Exhibit 1006, Abstract; *see also id.*, ¶ 6. Wu explains that "the subject or content of an email may be used repeatedly

in some occasions, and it is a waste of time to enter repeatedly the same information." *Id.*, ¶ 5. Accordingly, Wu discloses a system and a method of producing email by using predetermined templates. *See id.*, ¶¶ 7, 8, 17–32, Figures 1 and 2. Wu explains that "common information entered for an email is pre-stored in different databases." *Id.*, ¶ 9.

IX. SPECIFIC GROUNDS FOR UNPATENTABILITY

61. In my opinion, for the reasons noted below, claims 1–8 and 17–21 of the '520 patent would have been obvious in light of the prior art.

A. Ground 1: Claims 1–8 and 17–21 are Obvious over Houston and Garcia

1. Independent Claim 1

a) 1[preamble]: "An apparatus, comprising: at least one non-transitory memory storing instructions; and one or more processors in communication with the at least one non-transitory memory, wherein the one or more processors execute the instructions to:"

62. In my opinion, even if the preamble is limiting, Houston discloses it. For example, Houston discloses a host system that allows clients to share folders with one another. *See* Exhibit 1003, 1:53–2:2. In particular, Houston discloses a host system 110 that includes a metadata server 102, a block server 104, and a notification server 106. *Id.*, 2:59–62, Figure 1 (reproduced below). The host system is in communication with a plurality of clients (*e.g.*, clients 108a and 108b) via a network. *Id.*, Figure 1, 1:53–54, 6:5–6 The host system 110 manages folder

synchronization requests from the clients, blocks of data exchanged between it and the clients, and notifications to the clients. *See id.*, 2:63–3:3. Houston makes clear that its disclosed operations are performed by an "*apparatus*," which may comprise a computer storing a computer program. *Id.*, 9:5–9 ("The present invention also relates to an apparatus for performing the operations herein."). The computer program "may be stored on a non-transitory computer readable storage medium . . . suitable for storing electronic instructions." *Id.*, 9:10–18. Houston further discloses that the computer may include "a single processor" or "multiple processor designs." *Id.*, 9:18–21; *see also id.*, claim 5 ("A computer program product for synchronizing a folder across a computer network, the computer program product stored on a non-transitory computer readable medium and including instructions that when loaded into memory cause a computer processor to carry out the steps of: . . .").

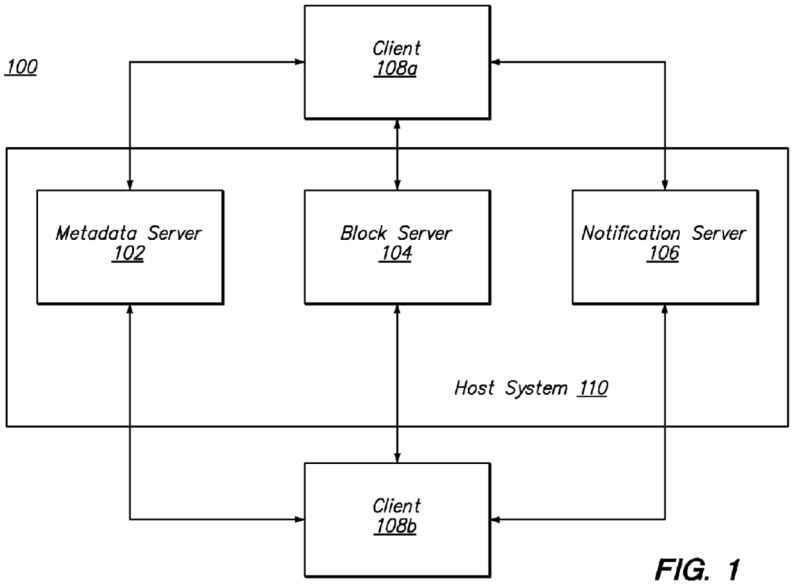

Exhibit 1003, Figure 1

63. Therefore, Houston discloses the preamble of claim 1.

b) 1[a]: "cause, at a first node, display of a first interface with a first user interface element;"

64. Houston discloses element 1[a]. Houston illustrates, in Figure 2 (reproduced below), a **user interface window 200** (*i.e.*, "*a first interface*") that a user can access "via a web interface" at a client 108 to share a folder. Exhibit 1003, 3:31–35. The **user interface window 200** includes a **text box** (*i.e.*, "*a first user interface element*") for the user to create a new folder to share.

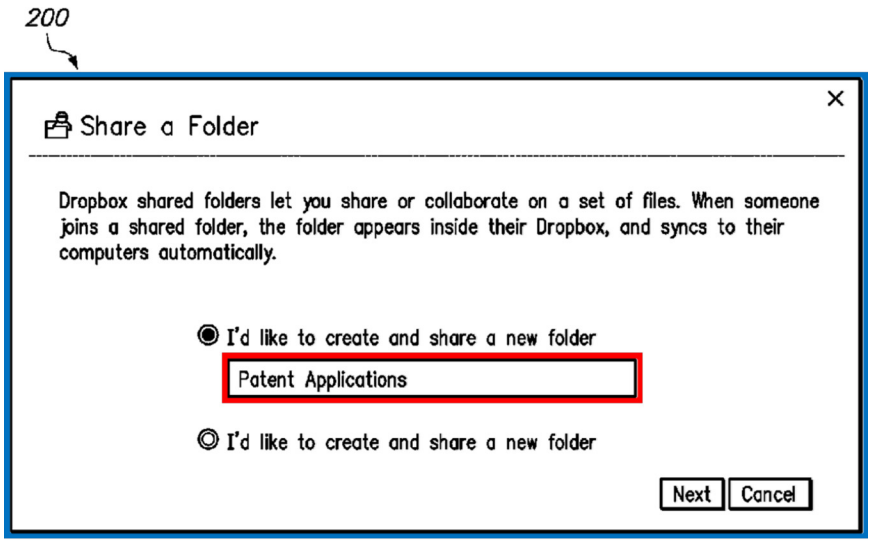

FIG. 2

Exhibit 1003, Figure 2 (annotated)

65. Alternatively, Houston discloses that the user can access a **user interface window 300** (*i.e.*, "*a first interface*") and select a folder to share from "from among existing folders" displayed in a **file structure** (*i.e.*, "*a first user interface element*"). *Id.* at 3:38-40, Figure 3 (reproduced below).

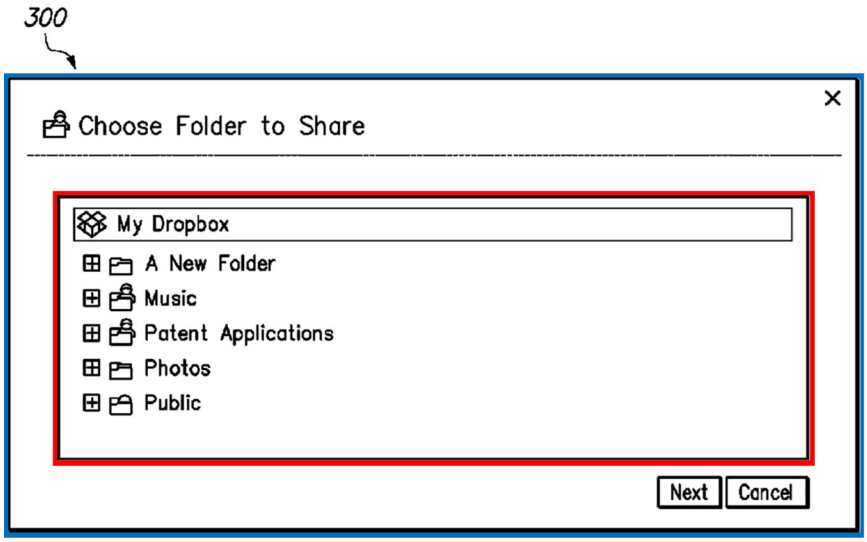

FIG. 3

Exhibit 1003, Figure 3 (annotated)

66. Each of the **user interface window 200** with the **text box** and the **user interface window 300** with the **file structure** is displayed via a web interface at a client—for example, **client 108a** (*i.e.*, "*a first node*")—in communication with the host system 110. *See id.*, 3:31–35, Figure 1. Because Houston explicitly discloses that the **user interface window 200** and the **user interface window 300** are displayed via "via a web interface," a POSITA would have understood that the host system 110 interacts with the client and "*causes display*" of the **user interface window 200** and the **user interface window 300** at the client. *See id.*

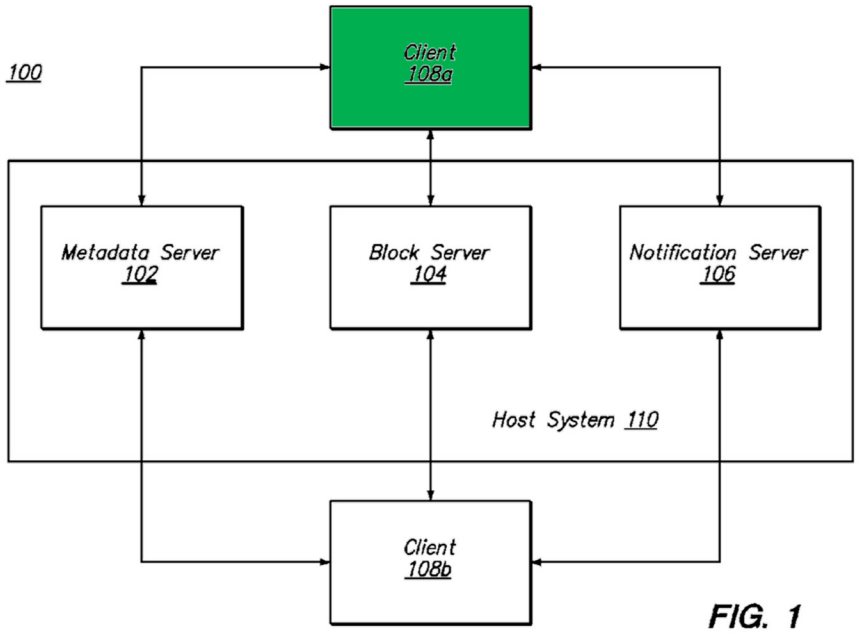

Exhibit 1003, Figure 1 (annotated)

67. Therefore, Houston discloses element 1[a].

c) 1[b]: "receive, from the first node via at least one network, an indication of at least one folder via the first user interface element, utilizing the first interface;"

68. Houston discloses element 1[b]. Referring to Figure 2, Houston explains that "the user has chosen to create a new folder called 'Patent Applications.'" Exhibit 1003, 3:37–38. The user (*e.g.*, at client 108a, "*the first node*") enters the folder name "Patent Applications" (*i.e.*, "*an indication of at least one folder*") utilizing the **text box** (*i.e.*, "*the first user interface element*") of the **user interface window 200** (*i.e.*, "*the first interface*"). Exhibit 1003, Figure 2.

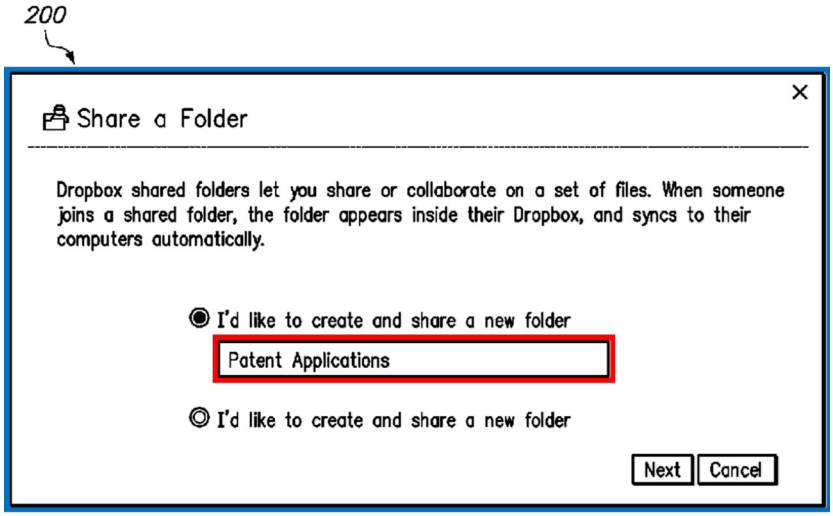

FIG. 2

Exhibit 1003, Figure 2 (annotated)

69. Alternatively, the user can choose an existing folder to share (*i.e.*, "*an indication of at least one folder*") utilizing the **file structure** (*i.e.*, "*the first user interface element*") of the **user interface window 300** (*i.e.*, "*the first interface*"). Exhibit 1003, Figure 3.

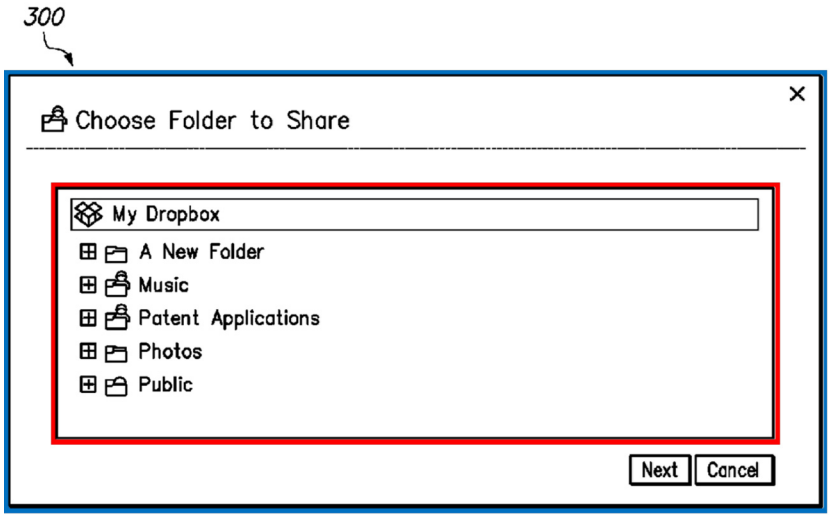

FIG. 3

Exhibit 1003, Figure 3 (annotated)

70. The "*indication of at least one folder*" is sent to and, accordingly, received by Houston's host system 110 from the client via the network (*e.g.*, shown in Houston's Figure 1) when (1) the user clicks on the "Next" button in Houston's Figure 2 or 3, or (2) the use clicks on a "Share folder" button in Houston's Figure 4. Either way, the host system 110 "*receive[s], from the first node via at least one network, an indication of at least one folder via the first user interface element, utilizing the first interface*."

71. Therefore, Houston discloses element 1[b].

d) 1[c]: "cause, at the first node, display of a second interface with a second user interface element;"

72. Houston discloses element 1[c]. Houston discloses that, "[o]nce the user has chosen or created the folder to be shared," the user is presented with a **user interface window 400** (*i.e.*, "*a second interface*") that allows the user to share the folder with other people. Exhibit 1003, 3:40–43, Figure 4 (reproduced below). The **user interface window 400** includes a **text box** (*i.e.*, "*a second user interface element*") for the user to enter the email address(es) of the other people.

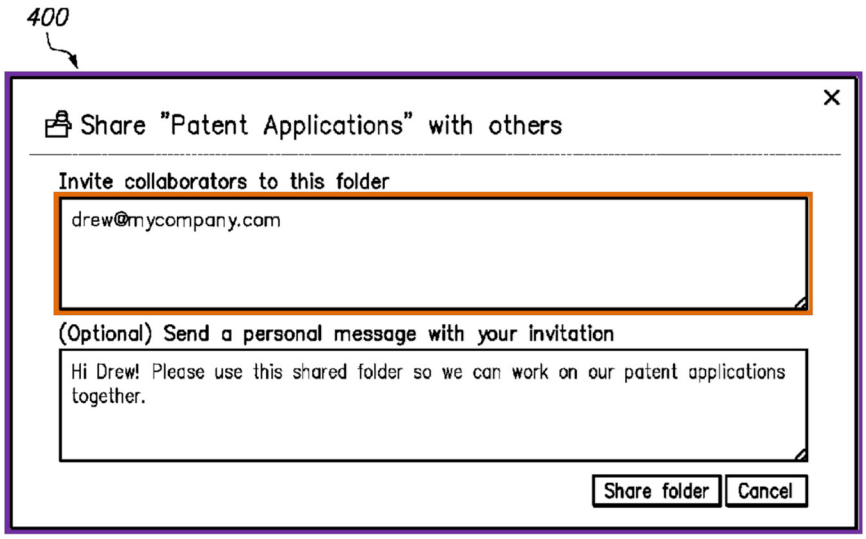

FIG. 4

Exhibit 1003, Figure 4 (annotated)

73. For at least the same reasons I explained above with respect to the user interface window 200 and the user interface window 300 in element 1[a], a POSITA would have understood that the host system 110 "*causes display*" of the **user interface window 400** at the client (*e.g.*, **client 108a**; *i.e.*, "*a first node*").

74. Therefore, Houston discloses element 1[c].

e) 1[d]: "receive, from the first node via the at least one network, indication associated with at least one email address via the second user interface element, utilizing the second interface;"

75. Houston discloses element 1[d]. As Houston illustrates in Figure 4, the user invites collaborators to the shared folder by entering an **email address** (*i.e.*, "*indication associated with at least one email address*") via the **text box** (*i.e.*, "*the second user interface element*") of the **user interface window 400** (*i.e.*, "*the second*

interface"). Exhibit 1003, Figure 4; *cf.* element 17[b] ("*the at least one object associated with at least one email address being the at least one email address*"). Houston's host system 110 "*receive[s]*" the email address to be used to notify the collaborator, from the user's client via the network, when the user clicks on the "Share folder" button.

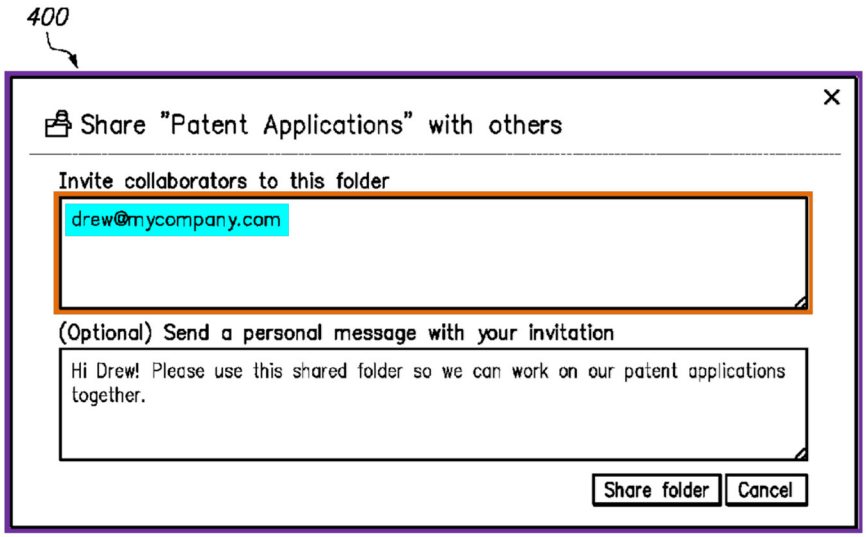

FIG. 4

Exhibit 1003, Figure 4 (annotated)

76. Therefore, Houston discloses element 1[d].

f) 1[e]: "receive, from the first node via the at least one network, an indication to share the at least one folder;"

77. Houston discloses element 1[e]. In Figure 4, Houston discloses a **"Share folder" button**. Exhibit 1003, Figure 4. A POSITA would have understood that the user clicking on the **"Share folder" button** would cause the host system 110 to receive instructions and information from the client, via the network, (1) to

share the folder chosen via user interface window 300 (*see* Houston's Figure 3) or (2) to create a new folder as specified in user interface window 200 (*see* "Patent Applications" in Houston's Figure 2) and share the created folder (*i.e.*, "*an indication to share the at least one folder*") with the collaborator whose email address was entered and received utilizing the text box of the user interface window 400. In other words, the host system 110 would "*receive, from the first node via the at least one network, an indication to share the at least one folder*."

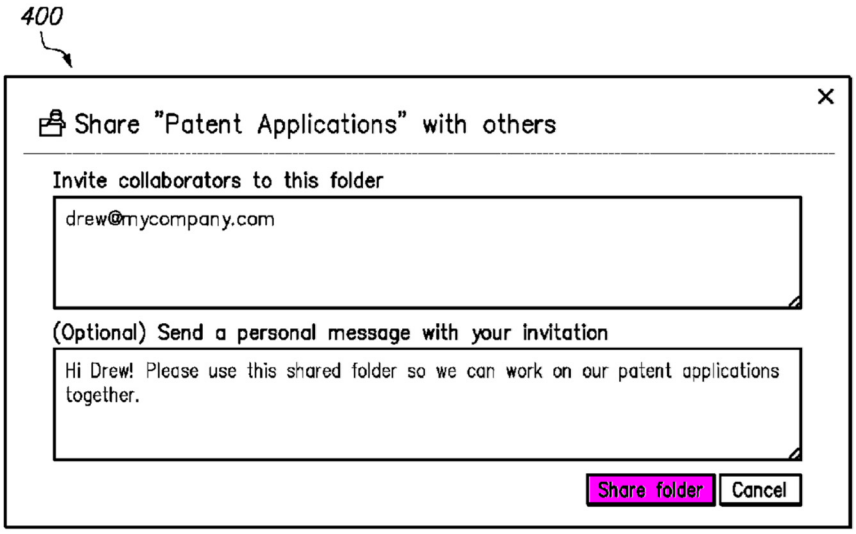

FIG. 4

Exhibit 1003, Figure 4 (annotated)

78. Therefore, Houston discloses element 1[e].

g) 1[f]: "based on the receipt of the indication of the at least one folder, the indicia associated with the at least one email address, and the indication to share the at least one folder via the at least one network; generate at least one email message identifying the at least one folder and including a reference to the at least one folder, without including at least one file in the at least one folder as an attachment of the at least one email message;"

h) 1[g]: "send, to a second node via the at least one network, the at least one email message, without including the at least one file in the at least one folder as an attachment of the at least one email message;"

79. Houston alone discloses or renders obvious, or in combination with Garcia, renders obvious elements 1[f] and 1[g].

i. Houston alone

80. ***First***, as I discussed above with respect to elements 1[b], 1[d], and 1[e], Houston discloses "*the receipt of the indication of the at least one folder, the indicia associated with the at least one email address, and the indication to share the at least one folder via the at least one network*."

81. ***Second***, Houston discloses that a plurality of clients may share folders with one another via the host system 110 over a network. Exhibit 1003, 3:23–24 ("any number of clients 108 may be sharing synchronized folders via host system 110"), 1:53–54 ("synchronization of folders shared among multiple clients over a network"). In particular, in addition to **client 108a** (*i.e.*, "*the first node*"), Houston refers to a **client 108b** (*i.e.*, "*a second node*"). *Id.*, Figure 1, 3:22–23.

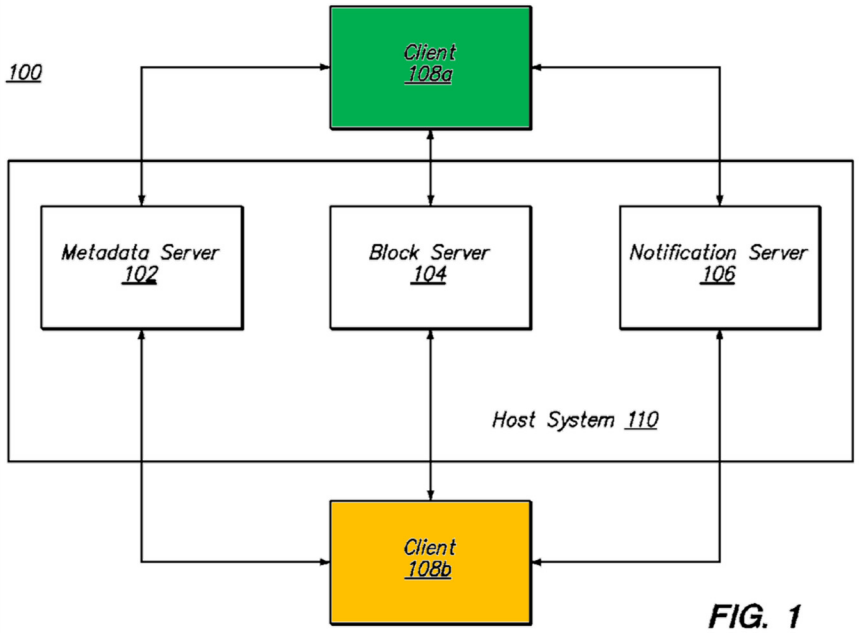

Exhibit 1003, Figure 1 (annotated)

82. ***Third***, because Houston's host system 110 receives, from a first client, information about a folder to share (*e.g.*, Figures 2 and 3) and a collaborator's email address (*e.g.*, "drew@mycompany.com" in Figure 4) and because Houston explicitly discloses that "the user can **invite** those people with whom he would like to share the folder" via the user interface window 400, a POSITA would have understood that Houston discloses its host system 110—particularly its notification server 106—would send an invitation to notify the collaborator about the folder being shared and how to access the folder by generating and sending an email message to the received email address (*i.e.*, "*generate at least one email message identifying the at least one folder and including a reference to the at least one folder*"; "*send, to a second node via the at least one network, the at least one email*

message"). Exhibit 1003, 3:40–43; *see also id*, 2:67–3:3 ("Notification server 106 provides updates to clients when a synchronized folder has been updated on the server, and provides those notifications to the clients."), 7:50–51 ("the user accepts the invitation to share the folder"). To the extent that the Patent Owner argues that Houston does not expressly disclose the claimed "email message," a POSITA would have appreciated from the same disclosure that an email would be the most practical implementation, rendering this limitation obvious.

83. ***Fourth***, Houston discloses that "sharing occurs at the folder level—that is, a folder and any files in that folder are shared among clients" and that "attaching files [is] cumbersome for many computer users," and therefore the email message would not include any file in the shared folder as an attachment (*i.e.*, "*without including at least one file in the at least one folder as an attachment of the at least one email message*"). Exhibit 1003, 3:5–10, 1:41–42; *see also id.*, 7:18–48.

ii. Combination of Houston and Garcia

84. To the extent that the Patent Owner contends that Houston alone does not render obvious elements 1[f] and 1[g], the combination of Houston and Garcia renders them obvious.

85. ***First***, Garcia discloses and illustrates in Figure 2 (reproduced below) "a plurality of client devices" (*e.g.*, home computer 230, work computer 240, mobile device 250) in communication with a synchronization server 220 via a network 210.

Exhibit 1004, 8:8–26. One of Garcia's client devices corresponds to a "*first node*" and another to a "*second node*." *Id.*, 8:21–24 ("These client devices 230, 240, 250 . . . may belong to different users who have shared access to objects with one another.").

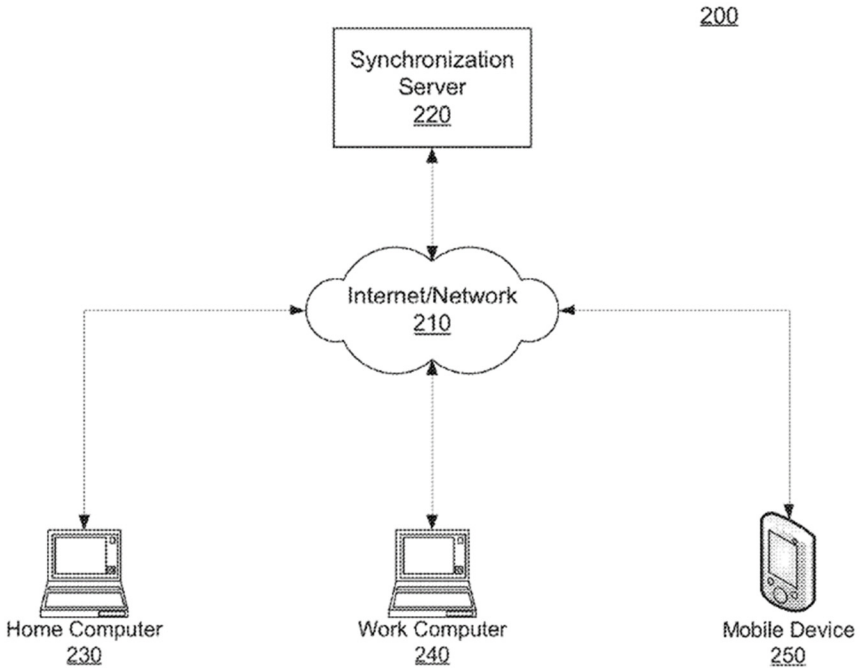

Exhibit 1004, Figure 2

86. ***Second***, Garcia teaches that a client can share a folder with another client by generating and sending an email with a link to and identifying the shared folder, instead of an attachment. *See* Exhibit 1004, 10:32–34 ("The Web link tab 335 includes a list of folders or files that have been shared as a link with other users."), 10:38–40 ("[A]n icon 915 . . . places the link onto the clipboard to be copied into an email, text, or other transmittal."), 11:39–43, claim 3, Figures 9, 11–12.

iii. Motivation to Combine Houston and Garcia

87. A POSITA would have been motivated to modify Houston's host system according to and informed by Garcia's teachings such that Houston's host system would generate and send an email message with a link to and identifying a shared folder, without including an attachment, to the user with whom the folder is being shared.

88. ***First***, Houston and Garcia are from the same field of endeavor. Both references are directed to sharing folders. For example, Houston discloses a system that enables "[s]ynchronization of folders shared among multiple clients over a network." Exhibit 1003, Abstract, 1:53–54, Figure 1. Similarly, Garcia discloses a system "for enabling a user to synchronize, manage, and share files and folders across a plurality of client devices and a synchronization server." Exhibit 1004, Abstract, Figure 2. Thus, both references are directed to systems for sharing folders. Exhibit 1003, Abstract, 1:53–54; Exhibit 1004, Abstract. Accordingly, a POSITA would have understood that Garcia's teachings were directly applicable to Houston's system.

89. ***Second***, implementing Houston's host system to generate and send to the recipient an email message with a link to and identifying a shared folder, without including an attachment, as taught by Garcia, would have yielded expected, predictable results. Houston's host system receives, from a user at a first node,

information about a folder to share and an email address of the person with whom the folder is to be shared. Exhibit 1003, 3:31–43, Figures 2–4. Garcia discloses sharing a folder by generating and sending an email message with a link to and identifying the shared folder, without including an attachment. Exhibit 1004, 10:32–34, 11:39–43, claim 3, Figures 11–12. A POSITA would have combined the elements taught by Houston and Garcia—namely, the information received about the folder to be shared and the email address of the recipient as taught by Houston, and the email message to the recipient with a link to the shared folder without including an attachment as taught by Garcia—using known and routine computer programming principles to modify Houston's host system to generate and send an email message addressed to the recipient with a link to and identifying the shared folder, without including an attachment. This result would have been readily predictable and recognized by a POSITA.

90. ***Third***, a POSITA would have been motivated to combine the teachings of Houston and Garcia because generating and sending to the recipient an email message with a link to a shared folder and without including an attachment would have been obvious to try. There were only a few common and predictable sharing methods—*e.g.*, a physical drive or disc, an email with an attachment, the Web, and an email with a link. *See, e.g.*, Exhibit 1017 at 15 ("four methods occupied the top three spots for all personal groups: email, portable drives, the Web, and writeable

disks"), Table 3; Exhibit 1008, Figure 4 (illustrating an email with a link instead of an attachment). A POSITA would have pursued these potential solutions with a reasonable expectation of success because the analysis would have simply involved trying each one of these solutions.

91. ***Fourth***, the modification of Houston's system to include Garcia's teachings of generating and sending to the recipient an email message with a link to a shared folder and without including an attachment would have been readily implemented by simply modifying Houston's host system to generate and send such an email message using information already received from a user at a first node (*e.g.*, information about the folder to be shared and the recipient's email address). A POSITA would have implemented this modification using basic computer programming principles to modify Houston's host system to include Garcia's teachings of generating and sending to the recipient an email message which included a link to and identifying the shared folder, without including an attachment. This modification would have been well within the knowledge and skillset of a POSITA.

92. ***Fifth***, a POSITA would have been motivated to combine the teachings of Houston and Garcia because Garcia expressly disclosed a benefit of sending an email message with a link to a shared folder—"One benefit of sending links to folders or files to other users is it allows other users to access the folders or files

without requiring the other user to create an account in the synchronization system." Exhibit 1004, 10:40–43. A POSITA would have recognized other benefits—*e.g.*, sending links to shared folders via an email message, a well-known, widely-used, and convenient form of communication, without needing to attach and/or locally store such folders, which may be large. Houston itself identified the motivation for implementing a more elegant alternative to including attachments in email messages and storing multiple versions of the same document locally. *See,* Exhibit 1003, 1:41–45 ("Not only is attaching files cumbersome for many computer users, but where multiple iterations are involved, it is not difficult to end up with multiple versions of the same document, perhaps having the same or a similar file name, located in various places on a user's hard drive.").

iv. Reasonable Expectation of Success in Combining Houston and Garcia

93. A POSITA would also have had a reasonable expectation of success. The combination of Houston and Garcia represents a straight-forward implementation of generating and sending an email using information already received by Houston's host system, which a POSITA would have been familiar with and been able to implement. The reasons a POSITA would have expected success parallel those that provide motivation for this combination—including because both Houston and Garcia are directed to sharing folders. Exhibit 1003, Abstract, 1:53–

54; Exhibit 1004, Abstract; *see also* Section IX.A.1.h.iii. Moreover, a POSITA would have had a reasonable expectation of success in implementing Houston's system to generate and send an email message with a link to a shared folder, without including an attachment, because Houston's underlying specification was designed to be extensible and flexible in its application. *See* Exhibit 1003, 9:40–42 ("the disclosure of the present invention is intended to be illustrative, but not limiting, of the scope of the invention").

94. Accordingly, Houston alone or in view of Garcia renders obvious elements 1[f] and 1[g] and a POSITA would have been motivated to combine the teachings of these references and would have had a reasonable expectation of success. For these same reasons, a POSITA would have been motivated to combine the teachings of these references for all subsequently discussed claims and limitations in this Ground.

i) 1[h]: "based on the receipt of the indication of the at least one folder, the indicia associated with the at least one email address, and the indication to share the at least one folder via the at least one network; cause, utilizing code configured to be stored on a storage at the second node and further configured to cooperate with a file explorer interface, creation of a representation of the at least one folder in a location among one or more folders on the file explorer interface, where the storage at the second node does not store the at least one file when the creation of the representation of the at least one folder is caused;"

95. Houston alone or in combination with Garcia renders obvious element 1[h].

i. Houston alone

96. ***First***, as I discussed above with respect to elements 1[b], 1[d], and 1[e], Houston discloses "*the receipt of the indication of the at least one folder, the indicia associated with the at least one email address, and the indication to share the at least one folder via the at least one network*."

97. ***Second***, referring to Figure 10 (reproduced below), Houston discloses "an **interface** for interacting with shared folders" (*i.e.*, "*a file explorer interface*"). Exhibit 1003, 2:48–50; *see also id.*, 7:18–21 ("host system 110 performs namespace mapping functions, allowing users of system 100 to interact seamlessly with shared folders through their operating system's standard user interface"). Houston explains that the **"My Dropbox" folder 1002** is "a folder that is synchronized with host system 110" and "is displayed next to other conventional folders such as 'My

Meetings', 'My Received Files,' and others." *Id.*, 7:21–31. A POSITA would have understood that code (*e.g.*, software) that allows access to host system 110 is stored on the invited collaborator's client device (*i.e.*, "*the second node*") and cooperates with the **Figure 10 interface** in order to display the **"My Dropbox" folder 1002** among the other conventional folders, which are local to the client (*e.g.*, "My Meetings"). *See id.*, 3:43–45 ("Figure 10 and Figure 11 provide a view of how a folder can be shared **using software on client 108**."), 8:64–9:4 ("the process steps and instructions of the present invention could be embodied in software, firmware or hardware, and when embodied in **software, could be downloaded to reside on and be operated from different platforms** used by real time network operating systems").

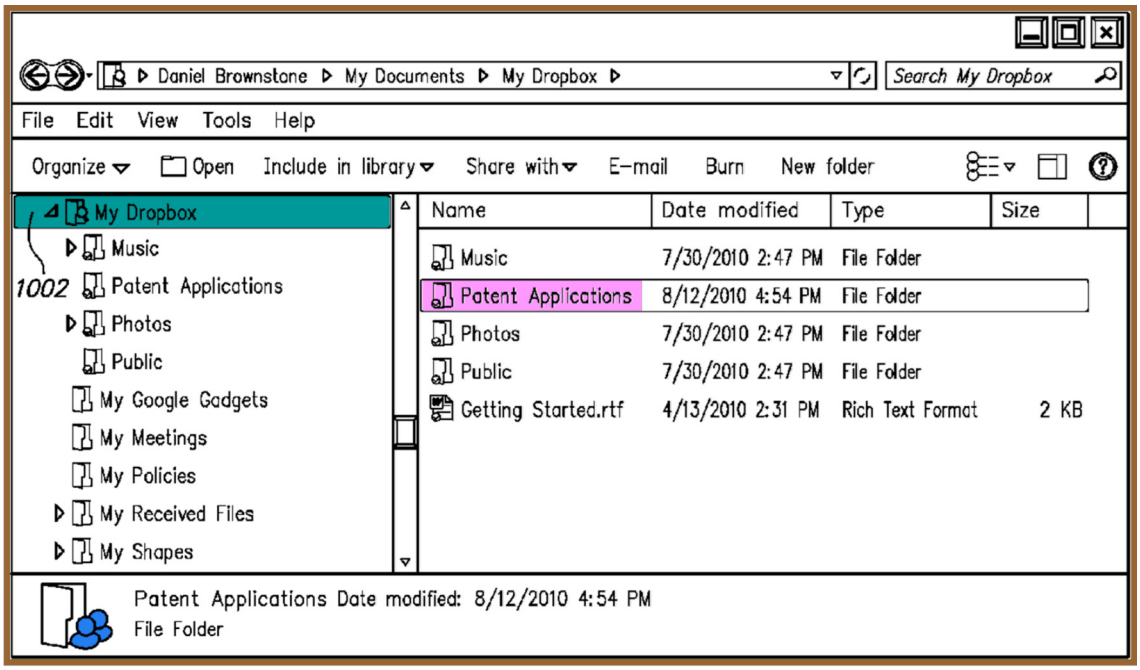

Exhibit 1003, Figure 10 (annotated)

98. ***Third***, because Houston discloses the **Figure 10 interface** as "an interface for interacting with shared folders," a POSITA would have understood that such an interface would be displayed on any of the clients 108—one where a user initiates sharing a folder (*e.g.*, client 108a, "*first node*") or another where a user is invited to access a shared folder (*e.g.*, client 108b, "*second node*"). In the latter case, for example, the **"Patent Applications" folder** would represent a shared folder displayed among other folders in the interface (*i.e.*, "*a representation of the at least one folder in a location among one or more folders on the file explorer interface*"). *See, e.g.*, Exhibit 1003, Figure 2 (illustrating a new folder to share named "Patent Applications"); *see also id.*, 7:49–58 (describing synchronizing a shared folder at an invited user's client device). Indeed, in Figure 10, Houston also illustrates the "Patent Applications" folder with a **"people" icon** to indicate that it is a shared folder. *See, e.g.*, Exhibit 1011 ("shared folders are shown with a blue 'people' icon").

99. ***Fourth***, as shown below, a close look at Houston's Figure 10 reveals **check mark overlays** on the icons for the folders and one file (*i.e.*, the check marks enclosed in circles on the lower left corner of the icons and annotated in green) in the "My Dropbox" folder 1002, but not on the conventional, local folders (*e.g.*, "My Meetings").

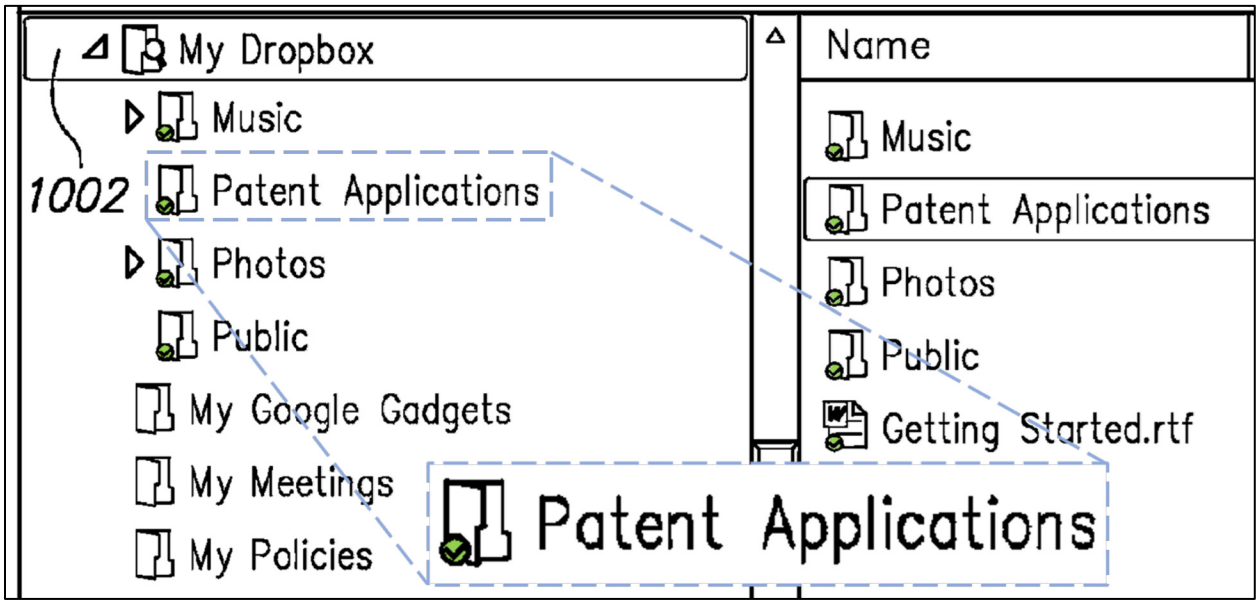

Exhibit 1003, Figure 10 (excerpted and annotated)

100. A POSITA would have understood that the **check mark overlays** indicate that these folders and file are synchronized with the host system 110. *See* Exhibit 1003 at 7:21–23 ("Assume a first user has a folder that is synchronized with host system 110. Referring to Figure 10, the "My Dropbox" folder 1002 is such a folder."); *see, e.g.*, Exhibit 1011 ("Files or folders that in sync with the cloud are represented by with green check mark . . ."). A POSITA would have further understood that no check mark overlay would show on these icons when they were first created in Houston's Figure 10 interface, in the moments before contents of these folders and file got synchronized with the host system 110. A POSITA would appreciate that the brief absence of check mark overlays indicated that the folders and file had not yet been synchronized with the host system 110. Thus, at the very

moment these icons were created (*i.e.*, "*when the creation of the representation of the at least one folder is caused*"), the client would not have stored any file within the folders, including the "Patent Applications" folder (*i.e.*, "*the storage at the second node does not store the at least one file*").

ii. Combination of Houston and Garcia

101. To the extent that the Patent Owner contends that Houston alone does not render element 1[h] obvious, the combination of Houston and Garcia renders it obvious.

102. ***First***, Garcia discloses and illustrates in Figure 21 (reproduced below) a synchronization system 2100 that includes a synchronization server 2120 in communication with a client device 2130 via an internet/network 2110. *See* Exhibit 1004, 6:36–67. Garcia describes that "all synchronized objects between a user's client devices and between users are stored on server 2120 and are accessible from any client device having access to the server 2120." *Id.*, 6:45–48; *see also id.*, 2:37–41 ("A user may be able to access, via [a] user interface, a virtual drive on the local device, where, via the virtual drive, a user is able to access items (e.g., files) in any object (e.g., folders), including objects located only on a remote device."). Garcia further explains that "[w]ithin client device 2130 is local synchronization client software 2140" (*i.e.*, "*code configured to be stored on a storage at the second node*"). *Id.*, 7:1–2. The local synchronization client software 2140 "functions to

synchronize client device 2130 with the synchronization server 2120." *Id.*, 7:2–3. Moreover, "[t]he local synchronization software 2140 is operatively coupled to a file system abstraction module 2150, within operating system 2155" of the client device 2130. *Id.*, 7:22–24. The operating system 2155 in turn is "connected to a **user interface 2170** (e.g., MAC FINDER or WINDOWS EXPLORER) which displays the physical drive user interface 2180 and the **virtual drive user interface 2190.**" *Id.*, 7:25–29. The **user interface 2170** thus corresponds to "*a file explorer interface*."

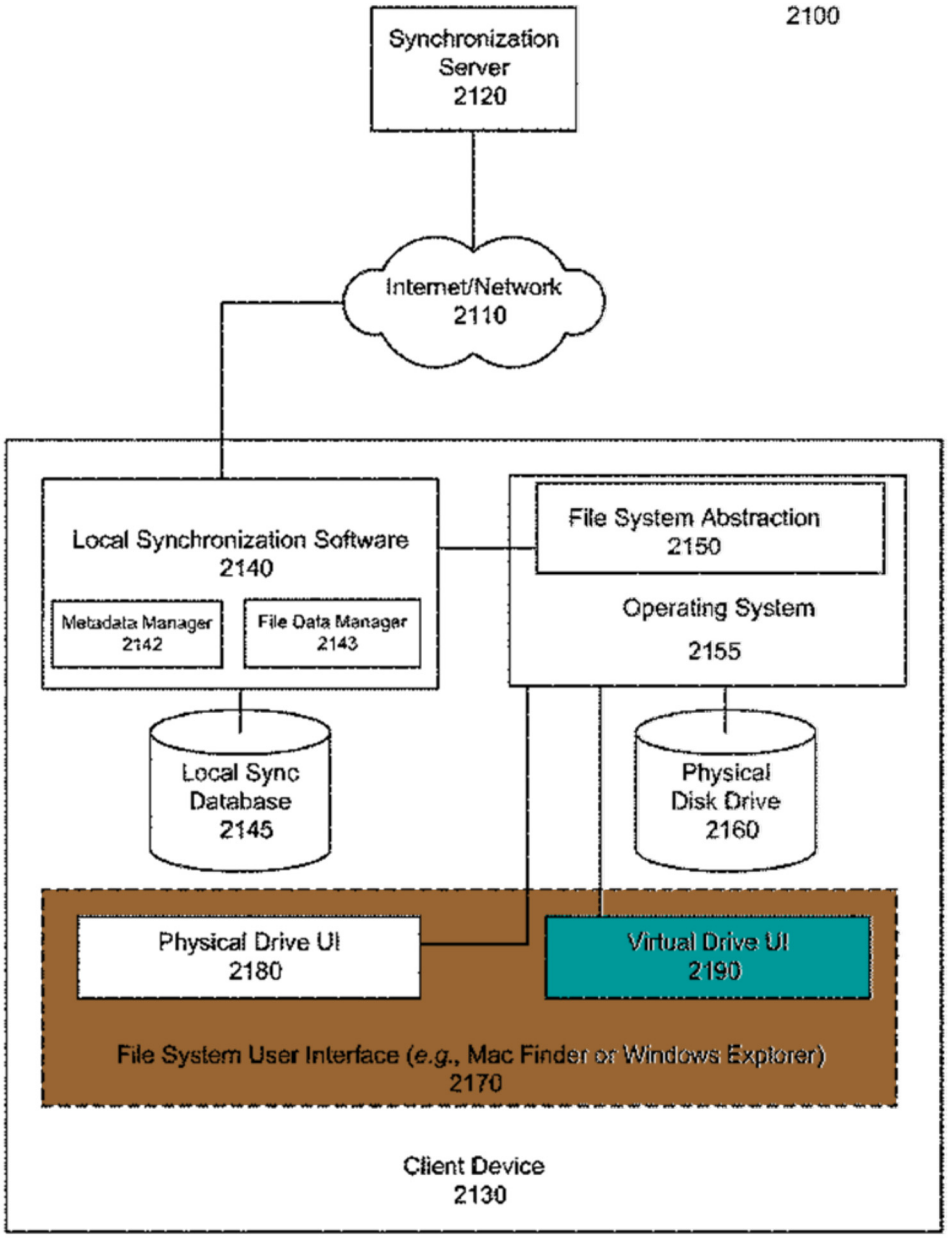

Exhibit 1004, Figure 21 (annotated)

103. ***Second***, consistent with its description of Figure 21, Garcia illustrates a **virtual drive 1010** displayed in an **interface** (*i.e.*, "*a file explorer interface*") in Figure 10 (reproduced below). As can be seen in Figure 10 and as Garcia explains,

"[t]he **virtual drive** appears as a separate disk drive on the user's local client device" in the **interface**. *Id.*, 5:27–30; Figure 20a (step 2015).

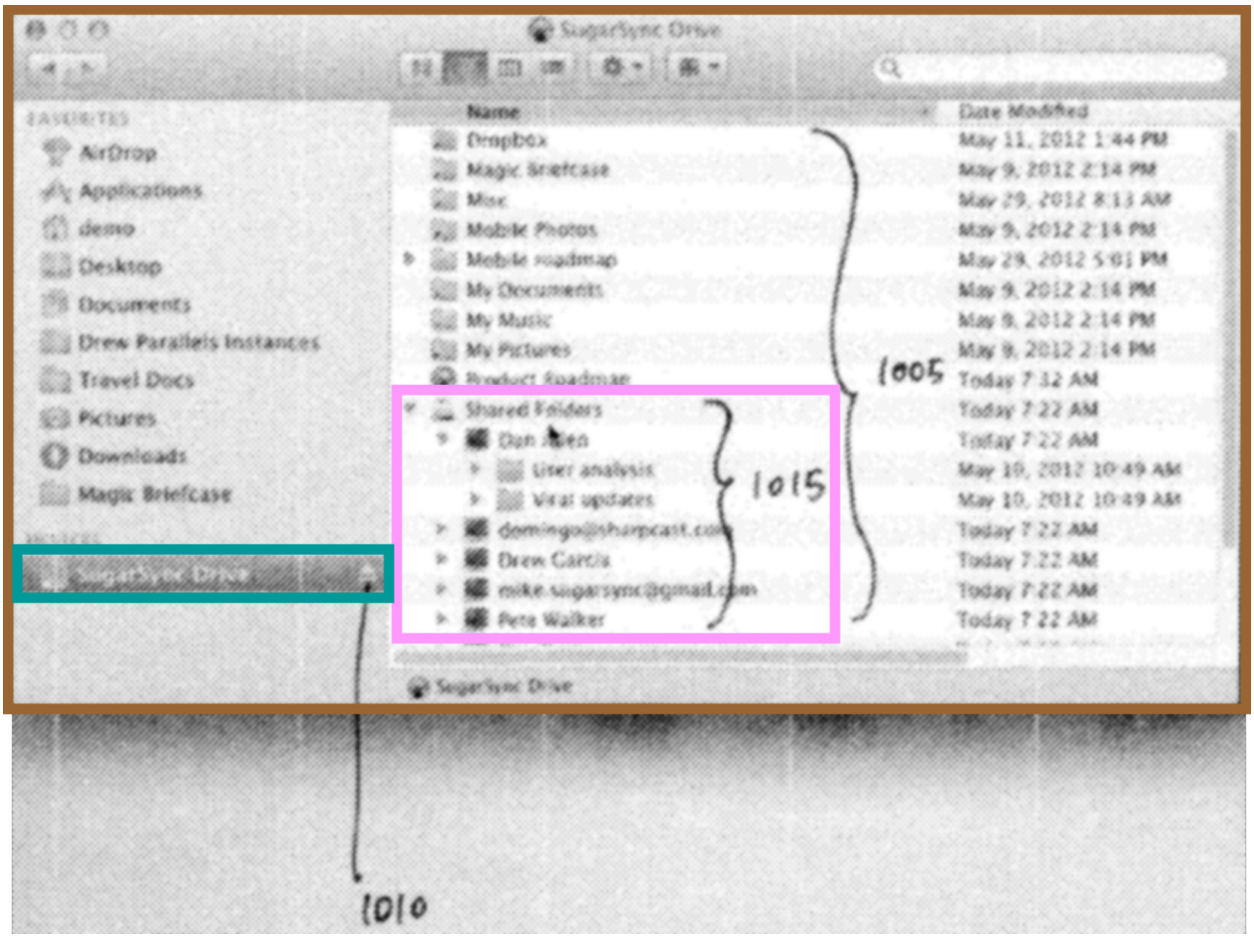

Exhibit 1004, Figure 10 (excerpted and annotated)

104. ***Third***, Garcia explains that "any file in folders 1005 in the synchronization system may be accessed through [the] **virtual drive 1010**" and that "[n]ot only are all of the user's folders that have been synchronized to or stored on server 220 accessible through the virtual drive, but also all of the **folders 1015 shared with the user by other users**" (*i.e.*, "*creation of a representation of the at least one folder in a location among one or more folders on the file explorer*

interface"). *Id.*, 10:59–63, 11:7–10; *see also id.*, 7:14 ("Physical disk drive 2160 stores the user's local files."), 7:26–28 ("user interface 2170 . . . displays the physical drive user interface 2180 and the virtual drive user interface 2190").

105. ***Fourth***, Garcia discloses that "[a] user is enabled to access and edit any item within any folder displayed in the virtual drive." Exhibit 1004, 5:32–34. Garcia goes on to explain that, "when a user double clicks on a file or item in a remote folder in the virtual drive, the server downloads its copy of the file/item to the local synchronization software client (i.e., the client currently being used by the user)." *Id.*, 5:35–39. In other words, when a remote folder—including a shared folder (*e.g.*, one of the **folders 1015**)—is displayed in the **virtual drive 1010**, any file in that folder is not downloaded to and is not stored in the storage of the client device until the user double clicks on the file (*i.e.*, "*the storage at the second node does not store the at least one file when the creation of the representation of the at least one folder is caused*").

iii. Motivation to Combine Houston and Garcia and Reasonable Expectation of Success in Combining Houston and Garcia

106. As I discussed above in Section IX.A.1.h.iii, a POSITA would have been motivated to modify Houston's system based on and informed by Garcia's teachings. A POSITA would have been further motivated to modify Houston's system such that, when an icon for a shared folder is created in Houston's Figure 10

interface at an invited user's client, the client does not store a file in the shared folder, as taught by Garcia. This would have provided the added benefit of allowing the invited user to have more control over what is downloaded from the shared folder and when, rather than letting the system download any file in the shared folder automatically onto the invited user's client from the host system 110. For instance, whether and when the invited user downloads a file in the shared folder would depend on whether the user's device has sufficient storage space or whether the file looks secure to the user or passes security checks. A POSITA would have implemented this modification using basic computer programming principles to modify Houston's host system to include Garcia's teachings and would have had a reasonable expectation of success. *See* Sections IX.A.1.h.iii and IX.A.1.h.iv.

107. Accordingly, Houston alone or in view of Garcia renders obvious element 1[h] and a POSITA would have been motivated to combine the teachings of these references and would have had a reasonable expectation of success. For these same reasons, a POSITA would have been motivated to combine the teachings of these references for all subsequently discussed claims and limitations in this Ground.

j) 1[i]: "detect, at the second node, an indication to open the at least one file in the at least one folder; and"

108. The combination of Houston and Garcia renders element 1[i] obvious. Houston does not explicitly disclose a collaborator providing an indication to open

a file in a shared folder (*e.g.*, the "Patent Applications" folder). However, Garcia discloses element 1[i]. In particular, Garcia discloses that, "when a user double clicks on a file or item in a remote folder in the virtual drive, the server downloads its copy of the file/item to the local synchronization software client (i.e., the client currently being used by the user)" (*i.e.*, "*detect, at the second node, an indication to open the at least one file in the at least one folder*"). Exhibit 1004, 5:35–39.

109. Therefore, Houston in view of Garcia renders obvious element 1[i].

k) 1[j]: "in response to detection of the indication to open the at least one file in the at least one folder, cause retrieval of the at least one file via the at least one network for permitting display of the at least one file at the second node."

110. The combination of Houston and Garcia renders element 1[j] obvious. Houston does not explicitly disclose a client of a collaborator retrieving a file in a shared folder (*e.g.*, the "Patent Applications" folder) in response to the collaborator providing an indication to open the file. However, Garcia discloses element 1[j]. For instance, Garcia discloses that, "when a user double clicks on a file or item in a remote folder in the virtual drive, the server downloads its copy of the file/item to the local synchronization software client (i.e., the client currently being used by the user)" and "[t]he user can then edit the file/item as desired" (*i.e.*, "*in response to detection of the indication to open the at least one file in the at least one folder, cause retrieval of the at least one file via the at least one network for permitting*

display of the at least one file at the second node"). Exhibit 1004, 5:35–40. A POSITA would have understood that, in order for the user to edit the file, the user's client device would display the file.

111. Therefore, Houston in view of Garcia renders obvious element 1[j].

112. For the above reasons, Houston in view of Garcia renders claim 1 obvious.

2. Dependent Claim 2

> ***"The apparatus of claim 1, wherein the apparatus is configured such that at least one of: the indicia associated with the email address is received separately from the indication to share the at least one folder; the indicia associated with the email address is not received separately from the indication to share the at least one folder; the apparatus is configured to control whether the storage at the second node stores or does not store the at least one file; the apparatus is configured to not control whether the storage at the second node stores or does not store the at least one file; the code is configured to control whether the storage at the second node stores or does not store the at least one file; or the code is configured to not control whether the storage at the second node stores or does not store the at least one file."***

113. Houston in view of Garcia renders obvious claim 1, as I discussed above. Houston discloses the additional limitation of claim 2. As I discussed above with respect to elements 1[d] and 1[e], Houston's host system 110 receives the **email address** (*i.e.*, "*indication associated with at least one email address*") to be used to notify the collaborator when the host system 110 receives an indication of the user

clicking on the **"Share folder" button** (*i.e.*, "*an indication to share the at least one folder*"), via the same user interface window 400. *See* Sections IX.A.1.e and IX.A.1.f; Exhibit 1003, Figure 4 (reproduced below). In other words, Houston discloses "*the indicia associated with the email address is not received separately from the indication to share the at least one folder*," as recited in the alternative by claim 2.

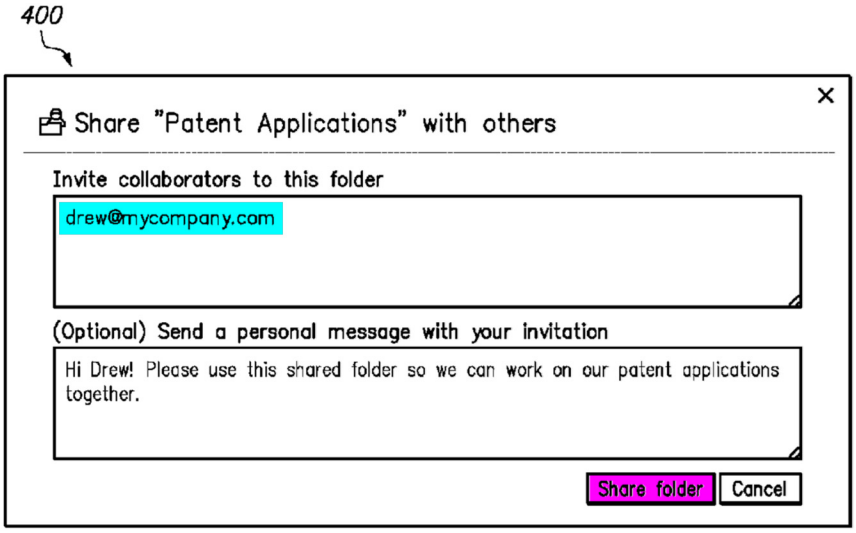

FIG. 4

Exhibit 1003, Figure 4 (annotated)

114. To the extent that the Patent Owner argues that Houston's host system 110 receives the **email address** and the indication of the user clicking on the **"Share folder" button** separately, then Houston discloses "*the indicia associated with the email address is received separately from the indication to share the at least one folder*," as recited in the alternative by claim 2.

115. Therefore, Houston in view of Garcia renders claim 2 obvious.

3. Dependent Claim 3

> ***"The apparatus of claim 1, wherein the apparatus is configured such that, when the creation of the representation of the at least one folder is caused, an additional representation on the file explorer interface is caused to be created that represents the at least one file, where the file explorer interface is of a file explorer client application and not a web page."***

116. Houston in view of Garcia renders obvious claim 1, as I discussed above. The combination of Houston and Garcia discloses the additional limitation of claim 3.

117. ***First***, Garcia discloses that, "when a user double clicks on a file or item in a remote folder in the virtual drive, the server downloads its copy of the file/item to the local synchronization software client (i.e., the client currently being used by the user)" and "[t]he user can then edit the file/item as desired." Exhibit 1004, 5:35–39. Thus, before the user double clicks on a file in a remote folder, which includes a shared folder (*e.g.*, one of the **folders 1015**), there is an icon for the file in the folder but the content of the file does not get downloaded until the user double click on the icon (*i.e.*, "*an additional representation on the file explorer interface is caused to be created that represents the at least one file*").

118. ***Second***, as I explained for element 1[h], Houston discloses displaying a representation of a shared folder (*e.g.*, the **"Patent Applications" folder**) in an

interface that is a "*file explorer interface*" and not a web page. *See* Exhibit 1003, Figure 10.

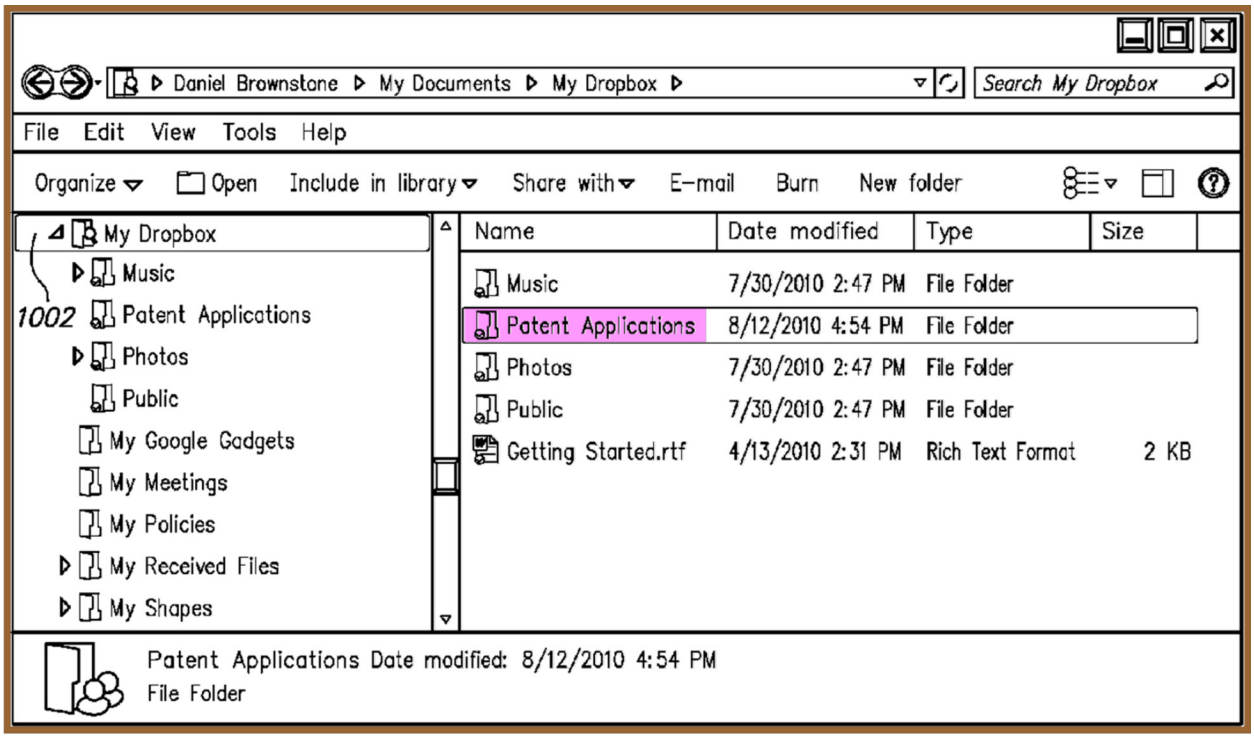

Exhibit 1003, Figure 10 (annotated)

119. As I also explained for element 1[h], Garcia discloses displaying a representation of a shared folder (*e.g.*, one of the **folders 1015**) in an **interface**. Exhibit 1004, Figure 10 (reproduced below). In particular, the **interface** corresponds to the application FINDER that is part of and runs on Macintosh operating systems. *See id.*, 7:25–29. ("The operating system 2155 is also connected to a user interface 2170 (e.g., MAC FINDER or WINDOWS EXPLORER) which displays the physical drive user interface 2180 and the virtual drive user interface

2190."). Thus, Garcia's **interface** is a "*file explorer interface*," which "*is of a file explorer client application and not a web page*."

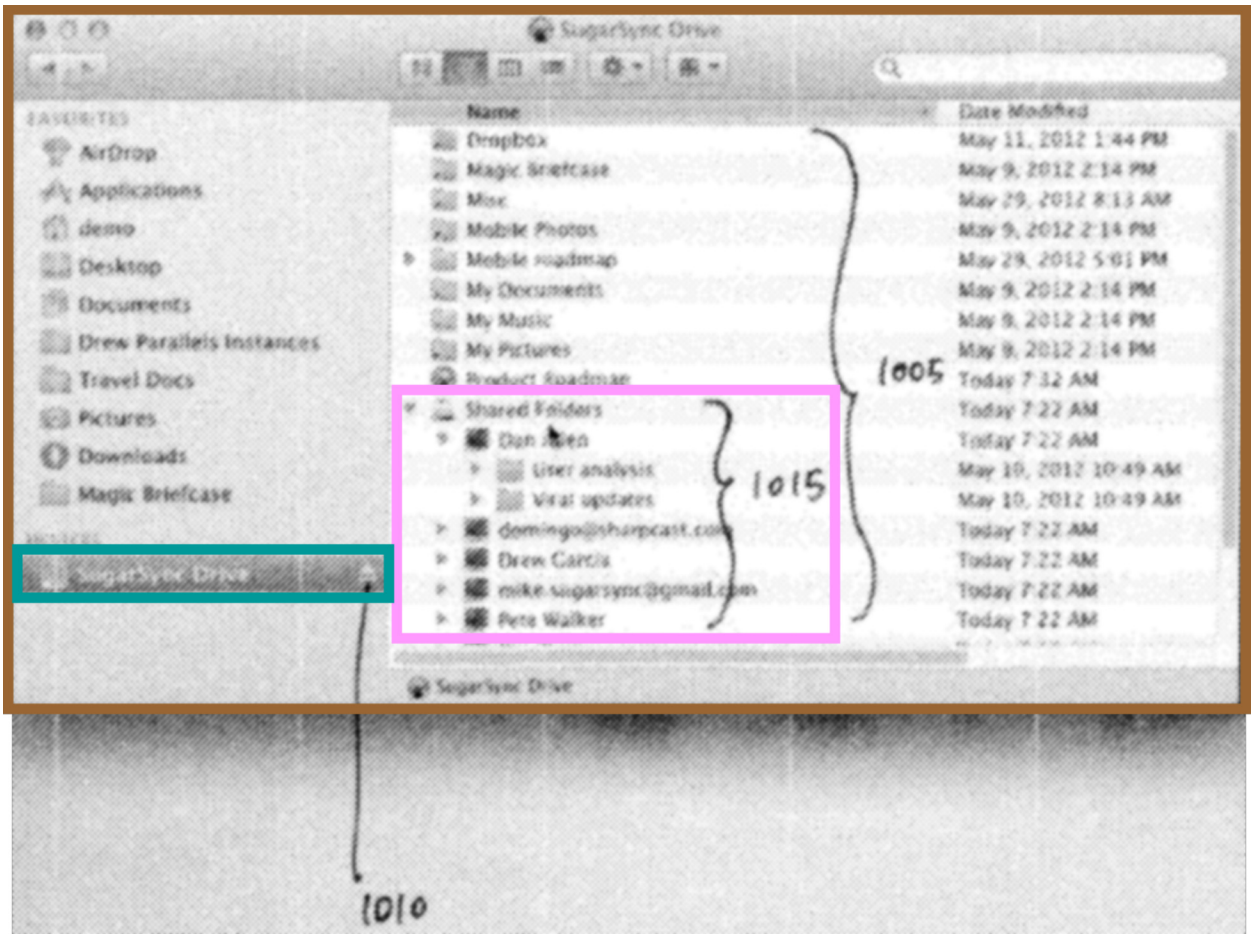

Exhibit 1004, Figure 10 (excerpted and annotated)

120. Therefore, Houston in view of Garcia renders claim 3 obvious.

4. Dependent Claim 4

> ***"The apparatus of claim 3, wherein the apparatus is configured such that the at least one folder and the at least one file are stored and synchronized utilizing at least one server, and wherein the apparatus is further configured such that the at least one email message is sent without including the at least one file as an attachment of the at least one email message, and the storage at the second node does not store the at least one file when the creation of the representation of the at least one folder is caused, both for avoiding the at least one file from having to be communicated to and stored at the second node until the at least one file is requested by a user at the second node using the additional representation."***

121. Houston in view of Garcia renders obvious claim 3, as I discussed above. The combination of Houston and Garcia discloses the additional limitation of claim 4.

122. ***First***, Houston discloses and illustrates in Figure 1 (reproduced below) "a host system and clients for maintaining synchronized shared folders." Exhibit 1003, 2:57–58. Houston further discloses that "[h]ost system 110 further includes a metadata server 102; a block server 104; and a notification server 106" (*i.e.*, "*at least one server*"). *Id.*, 2:60–62. Houston explains that "[m]etadata server 102 receives requests from clients to update the server's copy of synchronized folders and provides clients with a list of metadata for files being synchronized" and "[b]lock server 104 receives, stores, and serves blocks of data constituting synchronized files"

(*i.e.*, "*the at least one folder and the at least one file are stored and synchronized utilizing at least one server*"). *Id.*, 2:63–67.

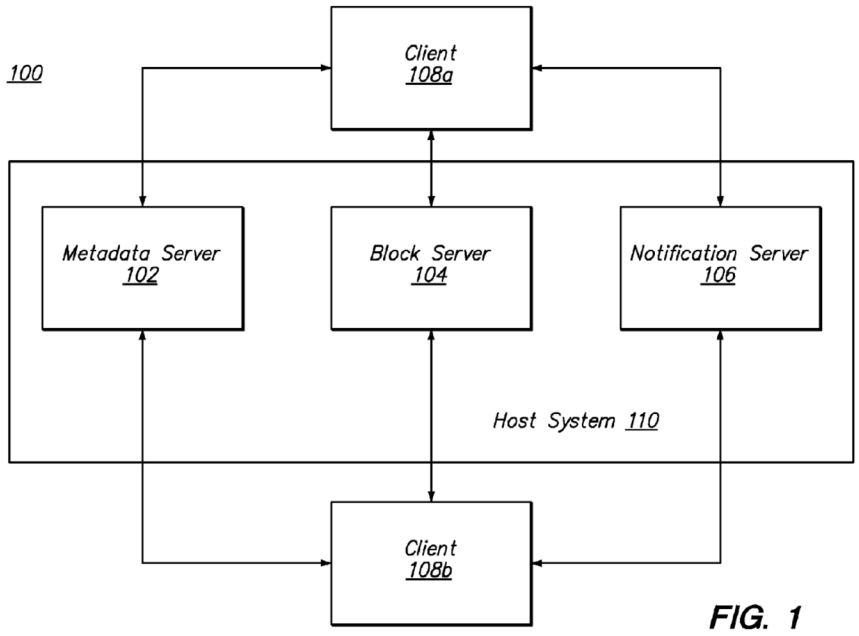

Exhibit 1003, Figure 1

123. Similarly, Garcia discloses "a system, method, and computer program for enabling a user to synchronize, manage, and share folders across a plurality of client devices and a synchronization server." Exhibit 1004, 3:13–16. In Figure 2 (reproduced below), Garcia illustrates client devices (*e.g.*, home computer 230, work computer 240, mobile device 250) in communication with a synchronization server 220. *Id.*, 8:8–26. Garcia explains that "[w]hen a user synchronizes an object to the synchronization server, the synchronization server stores a copy of the object." *Id.*, 8:29–31; *see also id.*, 6:45–48 ("all synchronized objects between a user's client devices and between users are stored on server 2120 and are accessible from any

client device having access to the server 2120"), 7:54–55 ("an object may be a folder or a file"). Therefore, according to Garcia, when a user shares a folder containing a file with another user, the folder and the file are stored and synchronized utilizing the synchronization server 220.

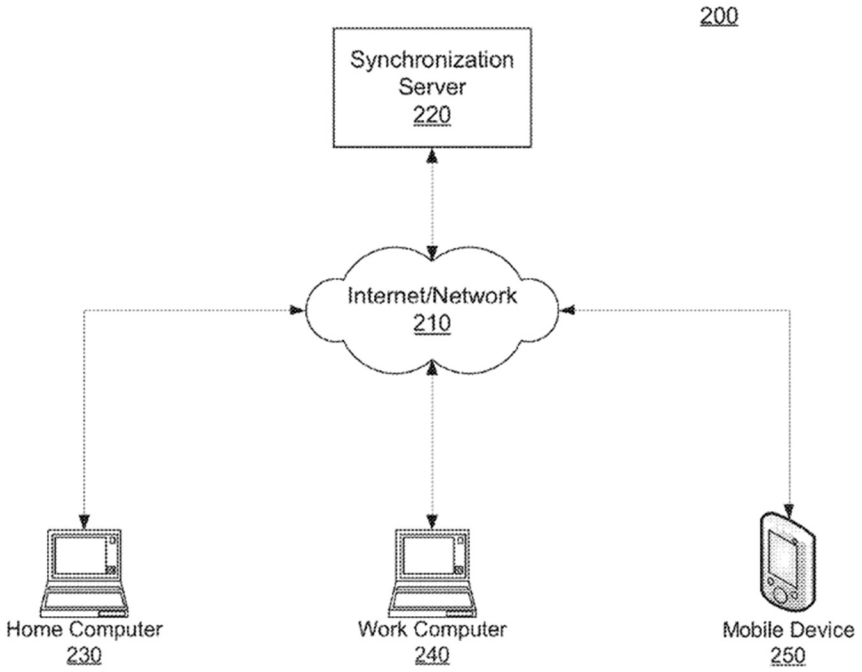

Exhibit 1004, Figure 2

124. ***Second***, as I discussed above for element 1[g], the combination of Houston and Garcia renders obvious "*the apparatus is further configured such that the at least one email message is sent without including the at least one file as an attachment of the at least one email message*." See Section IX.A.1.h. For at least the same reasons, the combination of Houston and Garcia renders obvious "*avoiding the at least one file from having to be communicated to . . . at the second node*."

125. ***Third***, as I discussed above for element 1[h], the combination of Houston and Garcia renders obvious "*the storage at the second node does not store the at least one file when the creation of the representation of the at least one folder is caused*." *See* Section IX.A.1.i. Indeed, Garcia discloses that "when a user double clicks on a file or item in a remote folder in the virtual drive, the server downloads its copy of the file/item to the local synchronization software client (i.e., the client currently being used by the user)" (*i.e.*, "*avoiding the at least one file from having to be . . . stored at the second node until the at least one file is requested by a user at the second node using the additional representation.*"). Exhibit 1004, 5:35–39.

126. Therefore, Houston in view of Garcia renders claim 4 obvious.

5. Dependent Claim 5

> ***"The apparatus of claim 1, wherein the apparatus is configured such that the file explorer interface displays representations of files in a file system of an operating system of the second node."***

127. Houston in view of Garcia renders obvious claim 1, as I discussed above. Garcia discloses the additional limitation of claim 5. For at least the same reasons I discussed above with respect to claim 3, Garcia discloses "*the file explorer interface displays representations of files*." Furthermore, Garcia discloses that the representations of files are in a file system of an operating system of a client device. According to Garcia, "[t]he local synchronization software 2140 is operatively coupled to a **file system** abstraction module 2150, within operating system 2155" of

the client device 2130 (*i.e.*, "*a file system of an operating system of the second node*"). Exhibit 1004, 7:22–24, Figure 21.

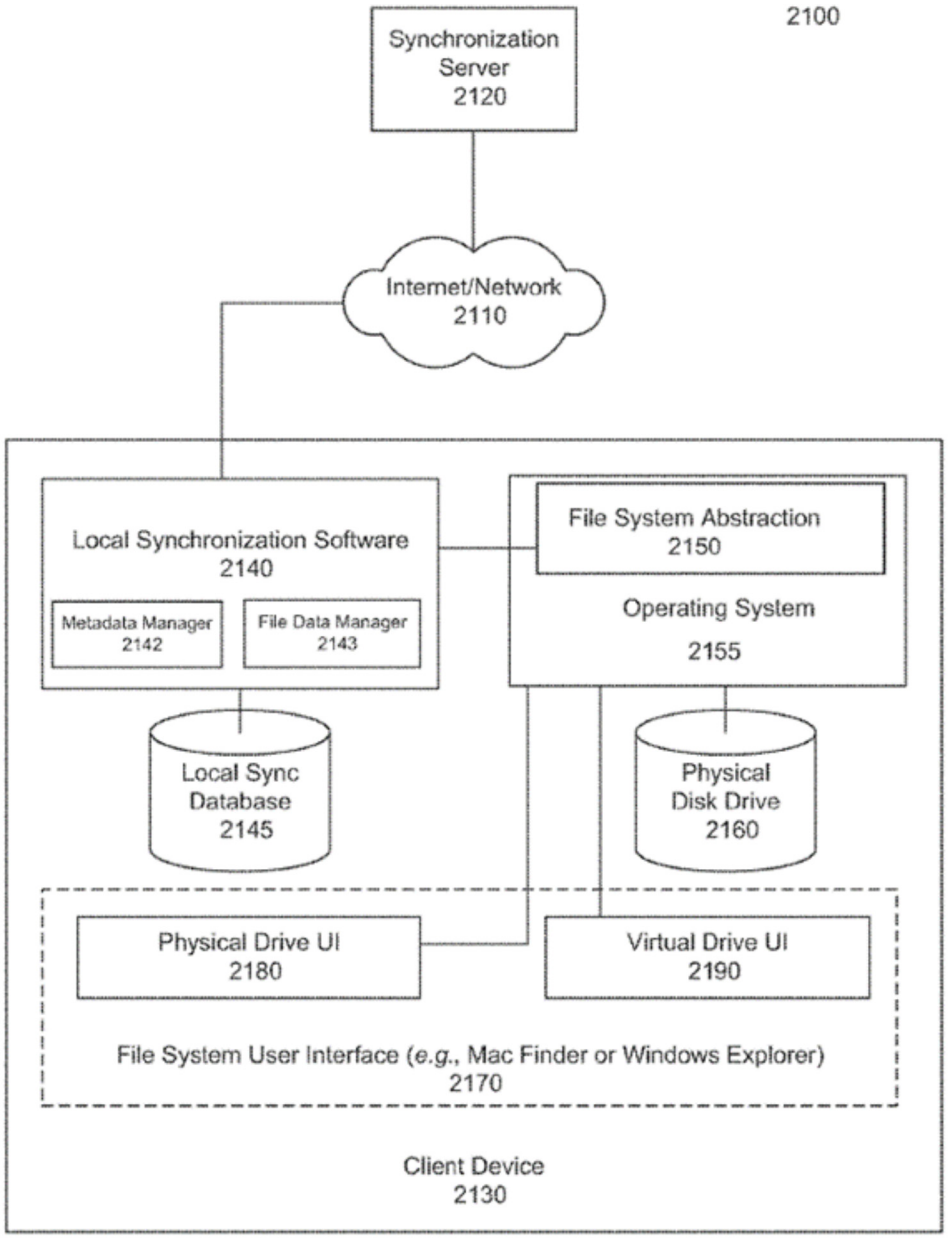

Exhibit 1004, Figure 21

128. Garcia also explains that "[t]he file system abstraction module 2150 functions to process and normalize the metadata related to the synchronized objects

so that the file system user interface module 2170 is able to display the virtual drive 2190." *Id.*, 7:29–33. Thus, Garcia discloses "*the file explorer interface displays representations of files in a file system of an operating system of the second node*."

129. Therefore, Houston in view of Garcia renders claim 5 obvious.

6. Dependent Claim 6

"The apparatus of claim 1, wherein the apparatus is configured such that the file explorer interface is part of an operating system of the second node."

130. Houston in view of Garcia renders obvious claim 1, as I discussed above. Garcia discloses the additional limitation of claim 6. As I explained above for claim 3, Garcia's **interface** in Figure 10 (reproduced below) corresponds to the application FINDER that is part of and runs on Macintosh operating systems (*i.e.*, "*the file explorer interface is part of an operating system*"). *See* Exhibit 1004, 7:25–29. ("The operating system 2155 is also connected to a user interface 2170 (e.g., MAC FINDER or WINDOWS EXPLORER) which displays the physical drive user interface 2180 and the virtual drive user interface 2190.").

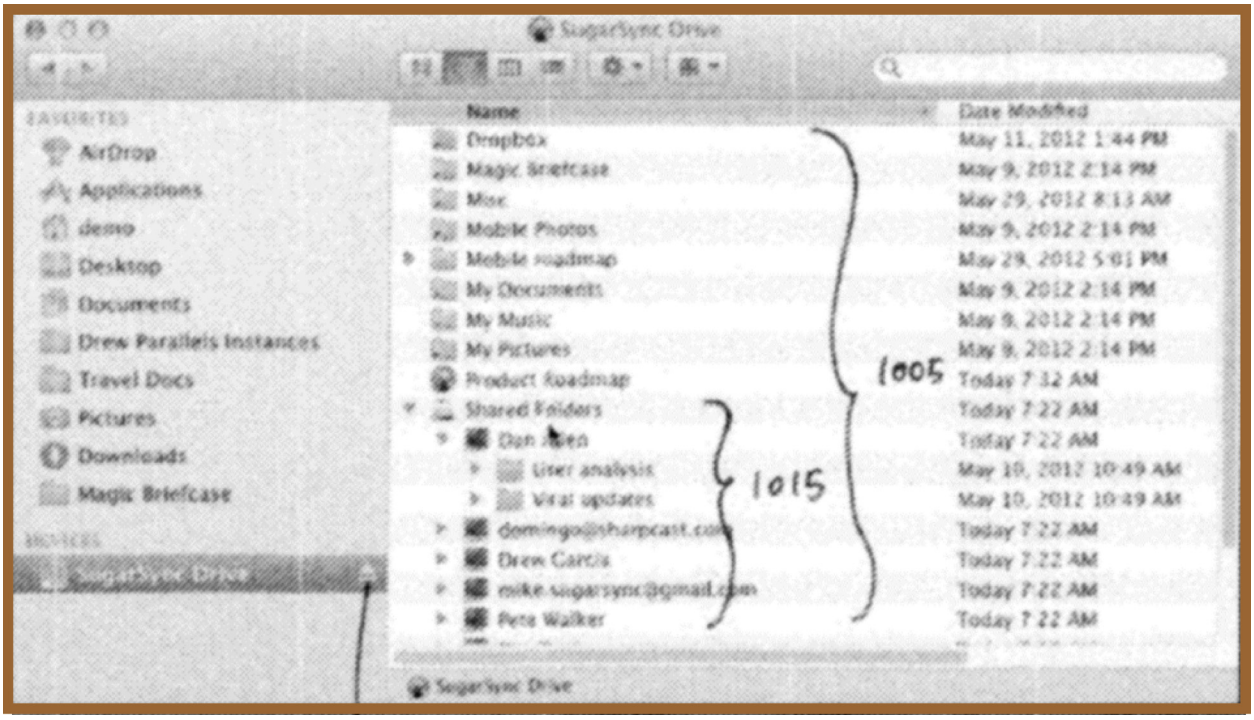

Exhibit 1004, Figure 10 (excerpted and annotated)

131. Therefore, Houston in view of Garcia renders claim 6 obvious.

7. Independent Claim 7

a) 7[preamble]: "A system, comprising: at least one non-transitory memory storing instructions; and one or more processors in communication with the at least one non-transitory memory, wherein the one or more processors execute the instructions to:"

132. In my opinion, even if the preamble is limiting, Houston discloses it for at least the same reasons I discussed above for element 1[preamble].

b) 7[a]: "display, at a first node, a first user interface element, a second user interface element, and a third user interface element;"

133. Houston discloses element 7[a]. For at least the same reasons I discussed above for elements 1[a] and 1[c], Houston discloses "*display, at a first node, a first user interface element [and] a second user interface element*." In addition, Houston discloses, in the same user interface window 400 that is displayed at client 108a (*i.e.*, "*first node*") and includes the **text box** (*i.e.*, "*second user interface element*") for email address(es), a **"Share folder" button** (*i.e.*, "*a third user interface element*"). Exhibit 1003, Figure 4 (reproduced below).

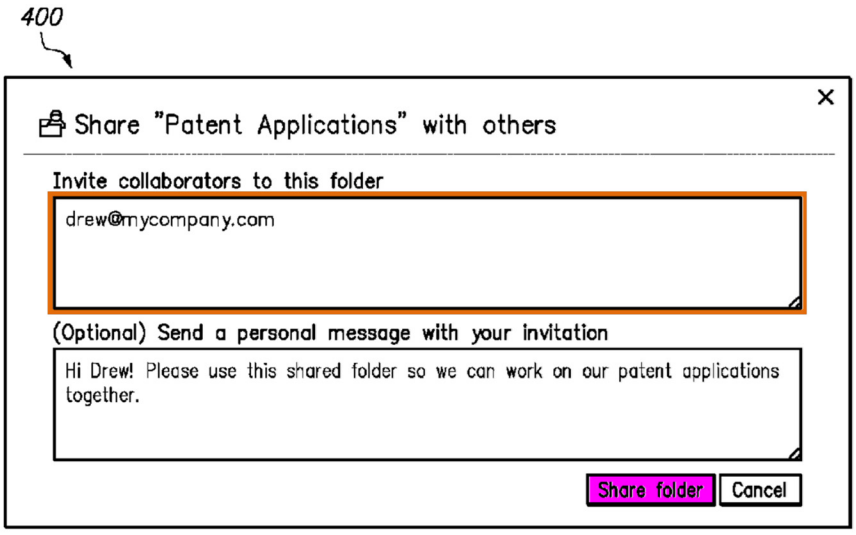

FIG. 4

Exhibit 1003, Figure 4 (annotated)

134. Therefore, Houston discloses element 7[a].

c) 7[b]: "receive, from the first node via at least one network, information associated with at least one data object via the first user interface element;"

135. Houston discloses element 7[b] for at least the same reasons I discussed above for element 1[b]. A data object includes a folder. *See, e.g.*, Exhibit 1004, 5:9–11 ("the objects are folders that may contain documents (e.g., EXCEL, POWERPOINT, or WORD documents), email/calendar items, video files, and/or photos"); *cf.* Exhibit 1001, 4:22–25 ("Exemplary data objects that may be requested via a data object identification request include a file, a program component, a data base record, video data, audio data, markup language, binary data, text data, an output of a service."), 27:43–48 ("allow a user to define a scope of a data object identification request by identifying whether a resulting data object identification request is to be applied to folders, files, and/or content of files in a file system and/or other data store of a receiving execution environment"), 43:35–44 ("a data object identification request may be based on a date, a time, a length of time, a file type, a database record key, content of the data object, a content type identifier, a format rule, a vocabulary, a role of a user, a security attribute, a location in a data store such as a file system, an attribute of an identified data object, a size, a task, a transaction, a state, a user, a group, a requester, a relationship including a requesting user and a responding user, a keyword, a tag, a folder, and/or a path portion of a data object identifier—to name a few examples").

d) 7[c]: "receive, from the first node via the at least one network, at least one object associated with at least one email address via the second user interface element;"

136. Houston discloses element 7[c] for at least the same reasons I discussed above for element 1[d]. *Cf.* element 17[b] ("*the at least one object associated with at least one email address being the at least one email address*").

e) 7[d]: "receive, from the first node via the at least one network, a message associated with a selection of the third user interface element;"

137. Houston discloses element 7[d] for at least the same reasons I discussed above for elements 1[e] and 7[a].

f) 7[e]: "based on the receipt of the information associated with the at least one data object, the at least one object associated with the at least one email address, and the message associated with the selection of the third user interface element via the at least one network; send, to a second node via the at least one network, at least one email message identifying the at least one data object and including a reference to the at least one data object, without including a file attachment with the at least one email message;"

138. Houston alone or in combination with Garcia renders obvious element 7[e] for at least the same reasons I discussed above for elements 1[f] and 1[g].

g) 7[f]: "based on the receipt of the information associated with the at least one data object, the at least one object associated with the at least one email address, and the message associated with the selection of the third user interface element via the at least one network; cause, utilizing code configured to be stored on a storage at the second node and further configured to cooperate with a file explorer interface of a client-based file explorer application, creation of a representation of the at least one data object in a location among one or more folders on the file explorer interface, where the storage at the second node does not store any file in the at least one data object when the creation of the representation of the at least one data object is caused."

139. Houston alone or in combination with Garcia renders obvious element 7[f] for at least the same reasons I discussed above for element 1[h]. In addition, as I discussed above for claims 3 and 6, Garcia's **interface** (*i.e.*, "*file explorer interface*") corresponds to the application FINDER that is part of and runs on Macintosh operating systems. *See* Exhibit 1004, Figure 10 (reproduced below), 7:25–29. ("The operating system 2155 is also connected to a user interface 2170 (e.g., MAC FINDER or WINDOWS EXPLORER) which displays the physical drive user interface 2180 and the virtual drive user interface 2190."). In other words, Garcia's **interface** corresponds to "*a file explorer interface of a client-based file explorer application*."

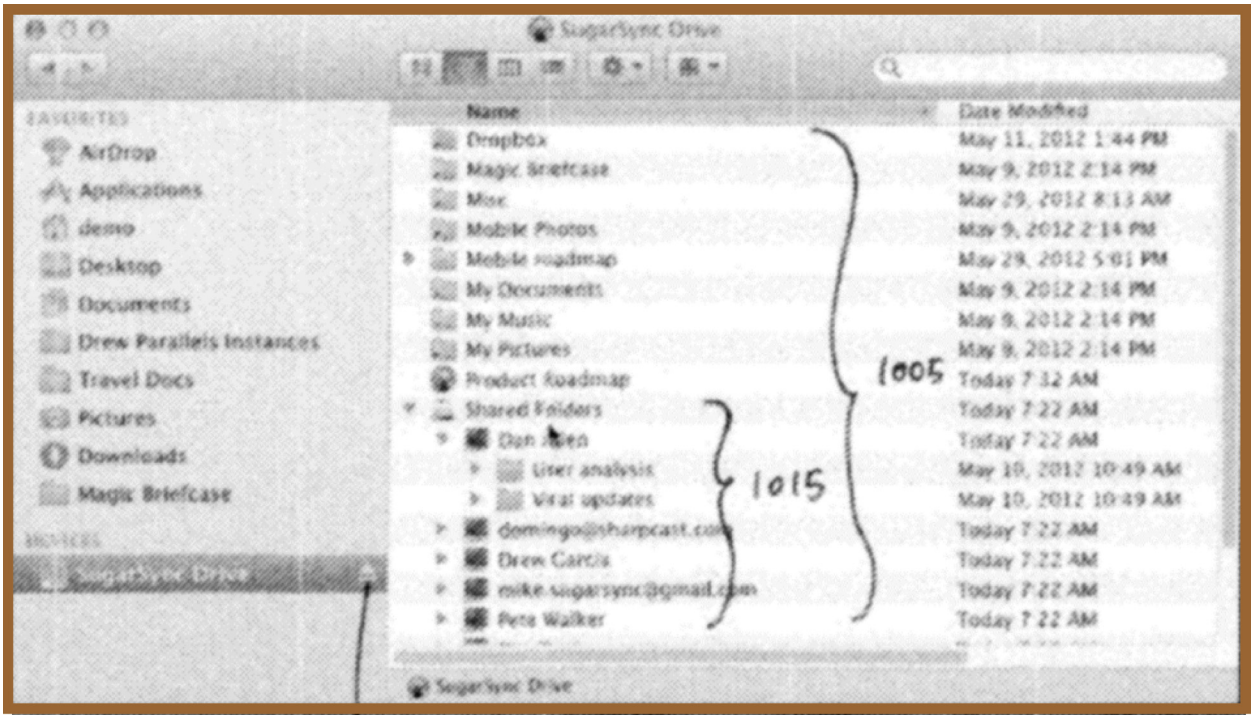

Exhibit 1004, Figure 10 (annotated)

140. Therefore, Houston in view of Garcia renders obvious element 7[f].

141. For the above reasons, Houston in view of Garcia renders claim 7 obvious.

8. Dependent Claim 8

> ***"The system of claim 7, wherein the system is configured such that the email message is sent without including the file attachment with the at least one email message, and the storage at the second node does not store any file when the creation of the representation of the at least one data object is caused, both for avoiding the at least one file from having to be communicated to and stored at the second node until the at least one file is requested by a user at the second node."***

142. Houston in view of Garcia renders obvious claim 7, as I discussed above. Houston discloses the additional limitation of claim 8 for at least the same reasons I discussed above for claim 4.

143. Thus, Houston in view of Garcia renders claim 8 obvious.

9. Independent Claim 17

a) 17[preamble]: "A method, comprising:"

144. In my opinion, even if the preamble is limiting, Houston discloses it. For example, Houston discloses "[a] method for synchronizing a shared folder over a network." Exhibit 1003, claim 1; *see also* Section IX.A.1.a.

> ***b) 17[a]: "causing, at a first node, display of: a first user interface element, for collecting information associated with at least one folder,"***

145. Houston discloses element 17[a] for at least the same reasons I discussed above for elements 1[a], 1[b], 7[a], and 7[b].

c) 17[b]: "[causing, at a first node, display of:] a second user interface element, for collecting at least one object associated with at least one email address, the at least one object associated with at least one email address being the at least one email address or an alias associated with the at least one email address, and"

146. Houston discloses element 17[b] for at least the same reasons I discussed above for elements 1[c], 1[d], 7[a], and 7[c].

d) 17[c]: "[causing, at a first node, display of:] a third user interface element, for detecting an indication of a selection thereof to cause an initiation of a sharing of the at least one folder;"

147. Houston discloses element 17[c] for at least the same reasons I discussed above for elements 1[e], 7[a], and 7[d].

e) 17[d]: "causing generation of at least one email, based on the information associated with the at least one folder, the at least one object associated with the at least one email address, and the detection of the indication of the selection of the third user interface element to cause the initiation of the sharing of the at least one folder, where the at least one email: identifies the information associated with the at least one folder, includes an Hypertext Transfer Protocol (HTTP) link, does not include a file attachment, for permitting avoidance of at least one file from being communicated to and stored at the second node until an initiation of the communication of the at least one file by a user of the second node is detected and the communication commences via at least one server that stores the at least one file, is at least partially pre-written, and is automatically caused to be received without requiring user involvement after the detection of the indication of the selection of the third user interface element to cause the initiation of the sharing of the at least one folder;"

148. Houston in combination with Garcia renders element 17[d] obvious for at least the same reasons I discussed above for elements 1[f] and 1[g] and claim 4. In addition, Garcia explicitly discloses sending an email with a link. Exhibit 1004, 10:32–34 ("The Web link tab 335 includes a list of folders or files that have been shared as a link with other users."), 10:38–40 ("[A]n icon 915 . . . places the link onto the clipboard to be copied into an email, text, or other transmittal."), 11:39–43, claim 3, Figures 11–12. A POSITA would have understood the link in Garcia's email to be a "*Hypertext Transfer Protocol (HTTP) link*," as was conventional. A POSITA would also have known that Garcia's email was partially pre-written (*e.g.*,

at least the "From" and "To" fields). Because neither Houston nor Garcia expressly requires user's approval for an email to be received, a POSITA would have understood the email to be automatically received by a receiving node (*e.g.*, Houston's client 108b) to which the email is addressed, without any user involvement, as was the norm.

149. Therefore, Houston in combination with Garcia renders element 17[d] obvious.

f) 17[e]: "receiving, from the second node and at at least one server, a signal for causing creation of a first representation of the at least one folder, in a location among one or more folders, that is stored at the at least one server and that is displayable via at least one web page;"

g) 17[f]: "causing, at the second node, receipt of the at least one web page, that results in display, at the second node and via the at least one web page, the first representation of the at least one folder that is stored at the at least one server; and"

150. The combination Houston and Garcia renders elements 17[e] and 17[f] obvious. For example, Houston discloses that the servers of its host system synchronize folders and store files therein. Exhibit 1003, 2:63-66 ("Metadata server 102 receives requests from clients to update the server's copy of synchronized folders and provides clients with a list of metadata for files being synchronized. Block server 104 receives, stores, and serves blocks of data constituting

synchronized files."). Houston further discloses that "a web interface allows access to shared folders from any computer with network access." Exhibit 1003, 2:10–12.

151. Similar to Houston, Garcia discloses that "[w]hen a user synchronizes an object to the synchronization server, the synchronization server stores a copy of the object." Exhibit 1004, 8:29–31; *see also id.*, 5:9–11 ("the objects are folders that may contain documents (e.g., EXCEL, POWERPOINT, or WORD documents), email/calendar items, video files, and/or photos"). Garcia discloses, and illustrates in Figure 13 (reproduced below), a web interface 1305 (*i.e.*, "*at least one web page*"), which includes a **Shared Folders tab 1335**. *Id.*, 11:44–50, Figure 13.

Exhibit 1004, Figure 13 (excerpted and annotated)

152. A POSITA would have understood that Garcia's synchronization server 220 (1) would have displayed the web interface 1305 at a client device (*e.g.*, home computer 230, work computer 240, mobile device 250), and (2) just like it displays "a list of folders 1345 from any of the synchronized client devices" in the My Cloud tab 1330, would have displayed a list of shared folders, including creating and displaying an icon for any new shared folder (*i.e.*, "*a first representation of the at*

least one folder"), in the **Shared Folders tab 1335**. *See, e.g.*, Exhibit 1004, Figure 2; Figure 10; 10:60–61 ("any file in folders 1005 in the synchronization system may be accessed through a virtual drive 1010"), 11:7–10 ("Not only are all of the user's folders that have been synchronized to or stored on server 220 accessible through the virtual drive, but also all of the folders 1015 shared with the user by other users.").

153. Therefore, Houston in combination with Garcia renders elements 17[e] and 17[f] obvious.

h) 17[g]: "causing, at the second node, receipt of code for storage at the second node and cooperation with a file explorer interface of a client-based file explorer application, for being utilized to: cause creation of a second representation of the at least one folder, in a location among one or more folders, that is stored at the second node and that is displayable via the file explorer interface of the client-based file explorer application."

154. Houston alone or in combination with Garcia renders obvious element 17[g] for at least the same reasons I discussed above for elements 1[h] and 7[f].

155. For the above reasons, Houston in view of Garcia renders claim 17 obvious.

10. Dependent Claim 18

> ***"The method of claim 17, wherein the email and the file explorer interface are displayed via separate interfaces for permitting access to the at least one folder via multiple different interfaces based on a detection of an indication that the HTTP link has been selected."***

156. Houston in view of Garcia renders obvious claim 17, as I discussed above. The combination of Houston and Garcia renders obvious the additional limitation of claim 18. As I discussed above for element 17[d], Houston alone or in combination with Garcia renders obvious "*at least one email . . . includes an Hypertext Transfer Protocol (HTTP) link*." As I discussed above for element 17[g], which refers to elements 1[h] and 7[f], Garcia's **interface** corresponds to a file explorer interface of a client-based file explorer application. *See* Exhibit 1004, Figure 10 (reproduced below). A POSITA would have known that clicking the HTTP link would have opened a web page in a web browser that is separate from a file explorer interface of a client-based file explorer application. A POSITA would have further understood that the user would be able to access the shared folder from the web page or the file explorer interface.

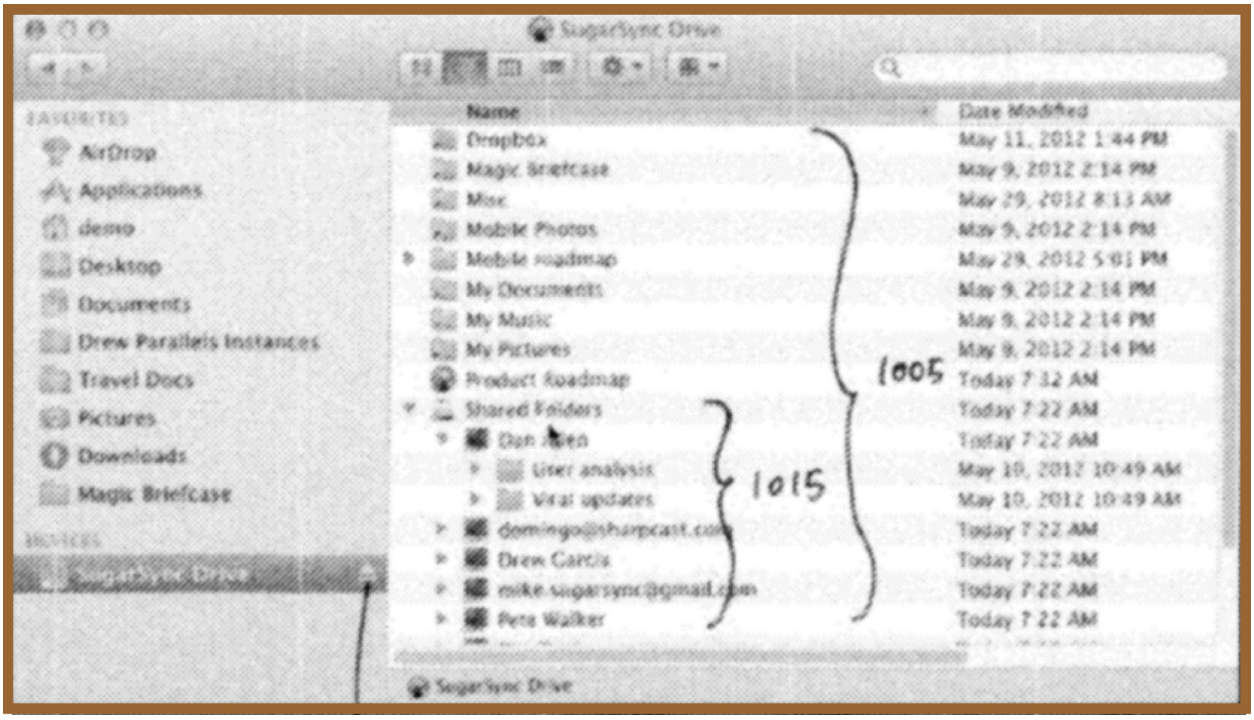

Exhibit 1004, Figure 10 (annotated)

157. Therefore, Houston in view of Garcia renders claim 18 obvious.

11. Dependent Claim 19

"The method of claim 17, wherein the file explorer interface of the client-based file explorer application is displayed via an interface that does not include a web page displayed via a web browser."

158. Houston in view of Garcia renders obvious claim 17, as I discussed above. Garcia discloses the additional limitation of claim 19. As I also discussed above and explained for claim 3, Garcia's **interface** corresponds to a file explorer interface of a client-based file explorer application, namely the application FINDER that is part of and runs on Macintosh operating systems. *See* Exhibit 1004, Figure 10 (reproduced below), 7:25–29. ("The operating system 2155 is also connected to

a user interface 2170 (e.g., MAC FINDER or WINDOWS EXPLORER) which displays the physical drive user interface 2180 and the virtual drive user interface 2190.”). FINDER is not a web browser. Moreover, as can been seen in Figure 10, Garcia’s **interface** does not include a web page displayed via a web browser.

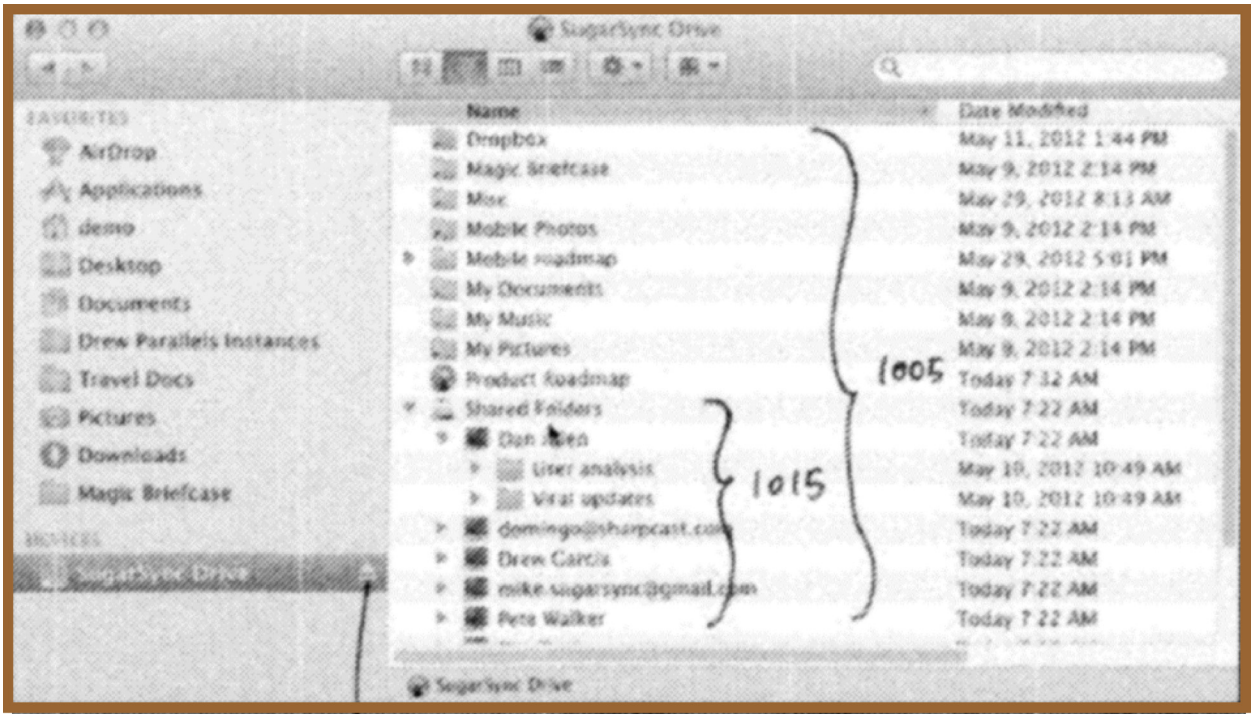

Exhibit 1004, Figure 10 (annotated)

159. Therefore, Houston in view of Garcia renders claim 19 obvious.

12. Dependent Claim 20

"The method of claim 17, wherein the file explorer interface is part of an operating system of the second node."

160. Houston in view of Garcia renders obvious claim 17, as I discussed above. Garcia discloses the additional limitation of claim 20 for at least the same reasons I discussed above for claim 6.

161. Therefore, Houston in view of Garcia renders claim 20 obvious.

13. Dependent Claim 21

"The method of claim 17, wherein at least one of: the first user interface element, the second user interface element, and the third user interface element are caused by sending a first communication; the receipt of the at least one web page is caused by sending a second communication; or the receipt of the code is caused by sending a third communication."

162. Houston in view of Garcia renders obvious claim 17, as I discussed above. Houston in combination with Garcia discloses or renders obvious the additional limitation of claim 21.

163. For example, as I discussed above for elements 17[e] and 17[f], a POSITA would have understood that Garcia's synchronization server 220 would have displayed the web interface 1305 (*i.e.*, "*the at least one web page*") at a client device (*e.g.*, home computer 230, work computer 240, mobile device 250). *See* Exhibit 1004, Figure 2 (reproduced below). Referring to Figure 2, a POSITA would

have understood that, for a client device to receive the web interface 1305, the synchronization server 220 would have sent a communication (*i.e.*, "*a second communication*") to the client.

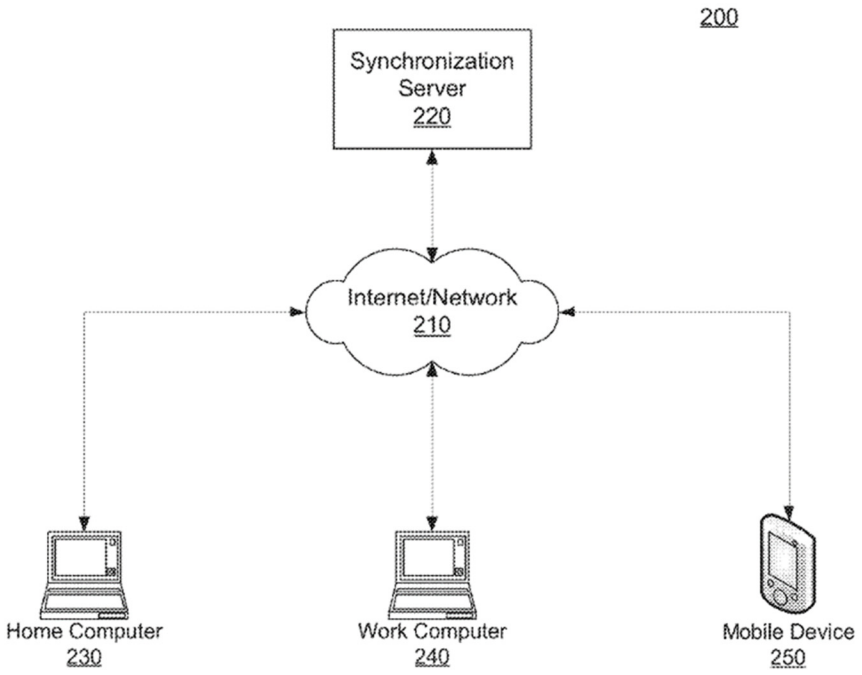

Exhibit 1004, Figure 2

164. Thus, Garcia discloses or at least renders obvious "*the receipt of the at least one web page is caused by sending a second communication*," as recited in the alternative by claim 21.

165. Therefore, Houston in view of Garcia renders claim 21 obvious.

B. Ground 2: Claims 1–8 and 17–21 are Obvious over Houston, Garcia, and Manzano

166. As I explained above in Section IX.A, the combination of Houston and Garcia renders obvious claims 1–8 and 17–21. However, to the extent that the Patent

Owner contends that this combination does not disclose, teach, suggest, or render obvious (1) "*the storage at the second node does not store the at least one file when the creation of the representation of the at least one folder is caused*" as recited by element 1[h], (2) "*an additional representation on the file explorer interface is caused to be created that represents the at least one file*" as recited by claim 3, (3) "*the storage at the second node does not store any file in the at least one data object when the creation of the representation of the at least one data object is caused*" as recited by element 7[f], and (4) "*a second representation of the at least one folder*" as recited by element 17[g] and to the extent that the Patent Owner further contends that a representation of a folder in a file explorer interface implies that at least one file in the folder is not downloaded when the representation of the folder is caused, the combination of Houston, Garcia, and Manzano renders these features obvious. I note that, unlike claims 1 and 7, claim 17 does not require that the second node does not store at least one file or any file in the at least folder when the creation of the second representation of the at least one folder or data object is caused.

167. For example, Manzano "relates generally to computer-implemented processes and, more specifically, to sharing of electronic files among electronic devices." Exhibit 1005, ¶ 2. Manzano discloses and illustrates in Figure 2 (reproduced below) a computer network system 200 comprising a server 230 linked to user devices 210, 280 via a network 260. *Id.*, ¶ 29. **User device 210** corresponds

to a "*first node*" and **user device 280** to a "*second node*." According to Manzano, "user devices 210, 280 include or are otherwise coupled to a computer screen or display 250, 290, respectively." *Id.*, ¶ 30.

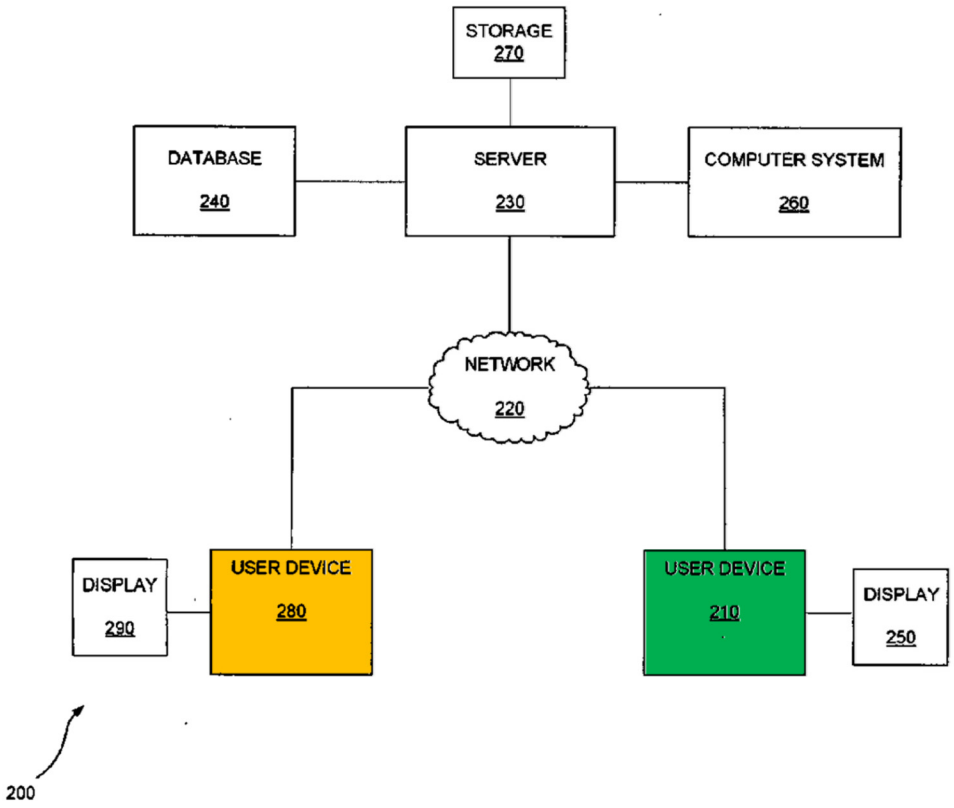

Exhibit 1005, Figure 2 (annotated)

168. Manzano teaches that "a file-transfer application" may be transferred from the server 230 to the user devices 210, 280 and that "[o]nce installed on the user devices 210, 280, the file-transfer application provides file-transfer clients 310, 320 executable by the user devices 210, 280, respectively." *Id.*, ¶ 32; *see also id.*, Figure 3 (reproduced below). Manzano discloses that "[a]lso included in the file-transfer application are user interfaces 350, 360 that are displayable on the displays

250, 290, respectively" and that "the interfaces 350, 360 correspond to respective electronic folders or windows into which files may be placed." *Id.*

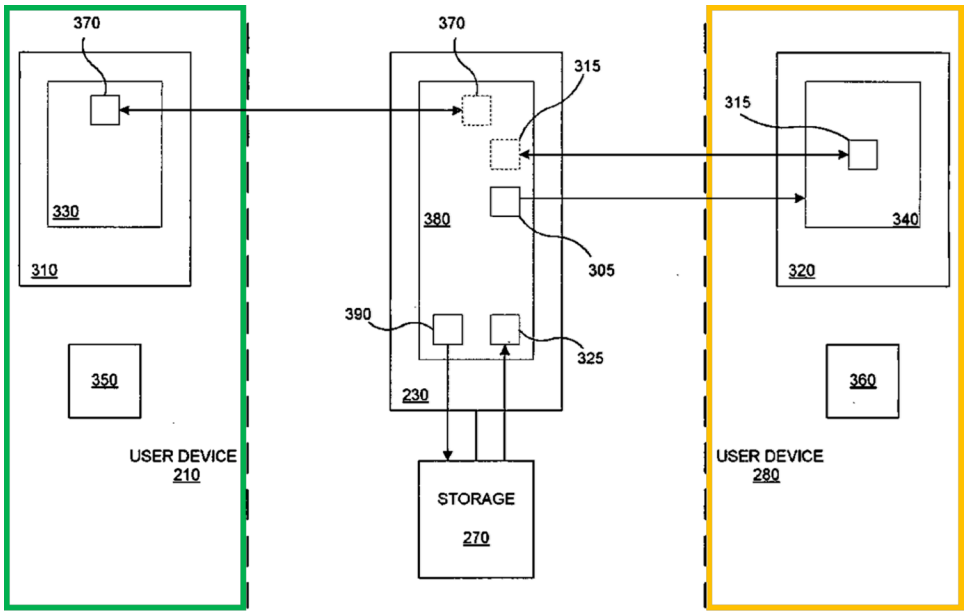

Exhibit 1005, Figure 3 (annotated)

169. Referring to Figure 6 (reproduced below), Manzano discloses that "file icons, such as the icon 600, in a workspace displayed on display device 290 can visually indicate whether a corresponding file (such as a file, the transfer of which may be as is discussed herein with reference to Figure 3) has been downloaded to or is otherwise stored on (i.e., is local to) . . . the user device 280." *Id.*, ¶ 50; *see also id.*, ¶ 52 ("'workspace' corresponds to a respective electronic folder"). Manzano explains that, if a shared file "is not local to **user device 280**, the icon 600 may be ghosted (i.e., semi-transparent) or otherwise be visually configured to indicate the non-local status of the file." *Id.*, ¶ 50.

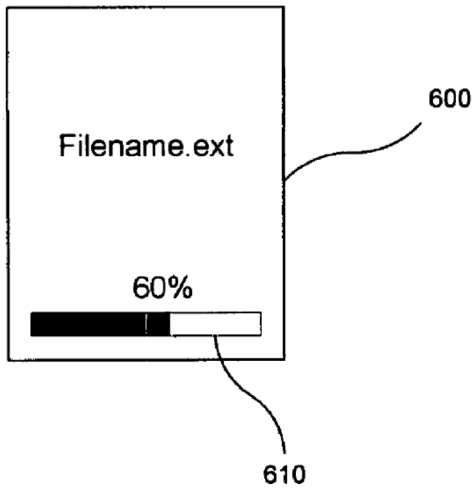

Exhibit 1005, Figure 6

170. Thereafter, at the request of a user of the **user device 280**, the file may be downloaded, after which the icon 600 will become "unghosted." *Id.*, ¶ 51 ("[A] user may right-click or otherwise indicate the icon 600, and from a resultant pop-up contextual menu (not shown), for example, that appears on the display 290, make a selection from the menu to request receipt or download of the associated file, at which time the icon 600 will become unghosted or will otherwise be visually configured to indicate the pending-local or local status of the file."). Thus, Manzano discloses that a storage at a second node does not store a shared file until the user at the second node expressly requests that the file be downloaded.

i. Motivation to Combine Houston, Garcia, and Manzano

171. A POSITA would have been motivated to further modify Houston's system as modified by Garcia's teachings (the "Houston-Garcia system") according to and informed by Manzano's teachings such that the Houston-Garcia system would

not store a file in a shared folder at a second node until the user at the second node requests the file be downloaded.

172. ***First***, Houston, Garcia, and Manzano are from the same field of endeavor. These references are directed to sharing electronic content among electronic devices over a network. For example, Houston discloses a system that enables "[s]ynchronization of folders shared among multiple clients over a network". Exhibit 1003, Abstract, 1:53–54, Figure 1. Similarly, Garcia discloses a system "for enabling a user to synchronize, manage, and share files and folders across a plurality of client devices and a synchronization server." Exhibit 1004, Abstract, Figure 2. Manzano relates "to sharing of electronic files among electronic devices" using a computer network system. Exhibit 1005, ¶¶ 2, 29, Figure 2. Thus, all three references are directed to sharing electronic content among electronic devices over a network. Exhibit 1003, Abstract, 1:53–54; Exhibit 1004, Abstract; Exhibit 1005, ¶¶ 2, 29. Accordingly, a POSITA would have understood that Manzano's teachings were directly applicable to the Houston-Garcia system.

173. ***Second***, implementing the Houston-Garcia system to not store a file in a shared folder at a second node until the user at the second node requests the file be downloaded, as taught by Manzano, would have yielded expected, predictable results. The Houston-Garcia system causes display of a shared folder at a second node. *See* Section IX.A.1.i. Manzano discloses that a storage at a second node does

not store a shared file until the user at the second node expressly requests that the file be downloaded. Exhibit 1005, ¶¶ 50–52, Figure 6. A POSITA would have combined the elements of the Houston-Garcia system and the elements taught by Manzano—namely, the shared folder of the Houston-Garcia system, and not downloading a shared file in a folder as taught by Manzano—using known and routine computer programming principles to modify the Houston-Garcia system to not store a file in a shared folder at a second node until the user at the second node requests the file be downloaded. This result would have been readily predictable and recognized by a POSITA.

174. ***Third***, a POSITA would have been motivated to combine the teachings of Manzano with the Houston-Garcia system because not storing a file in a shared folder at a second node until the user at the second node requests the file be downloaded would have been obvious to try. There were only a few known and predictable choices for presenting a file in a shared folder to a recipient—*e.g.*, (1) download the file automatically, or (2) download the file upon the recipient's request. A POSITA would have pursued these potential solutions with a reasonable expectation of success because the analysis would have simply involved trying each one of these solutions.

175. ***Fourth***, the modification of the Houston-Garcia system to include Manzano's teachings of not storing a file in a shared folder at a second node until

the user at the second node requests the file be downloaded would have been readily implemented by simply modifying the Houston-Garcia system to not store a file in a shared folder at a second node until the user at the second node requests the file be downloaded. A POSITA would have implemented this modification using basic computer programming principles to modify the Houston-Garcia system to include Manzano's teachings of not storing a file in a shared folder at a second node until the user at the second node requests for the file to be downloaded. This modification would have been well within the knowledge and skillset of a POSITA.

176. ***Fifth***, a POSITA would have recognized the benefit of not storing a file in a shared folder at a second node until the user at the second node requests the file be downloaded. For instance, such a feature would have allowed the recipient of the shared folder to have more control over what to download from the shared folder and when, rather than letting the system download any file in the shared folder automatically onto the recipient's device. For example, whether and when the invited user downloads a file in the shared folder would depend on whether the user's device has sufficient storage space or whether the file looks secure to the user or passes security checks.

ii. Reasonable Expectation of Success in Combining Houston, Garcia, and Manzano

177. A POSITA would also have had a reasonable expectation of success. The combination of the Houston-Garcia system and Manzano represents a straight-forward implementation of not storing a file in a shared folder at a second node until the user at the second node requests the file be downloaded, which a POSITA would have been familiar with and been able to implement. The reasons a POSITA would have expected success parallel those that provide motivation for this combination—including because Houston, Garcia, and Manzano are all directed to sharing electronic content among electronic devices over a network. Exhibit 1003, Abstract, 1:53–54; Exhibit 1004, Abstract; Exhibit 1005, ¶¶ 2, 29; *see also* Section IX.B.i. Moreover, a POSITA would have had a reasonable expectation of success in implementing in the Houston-Garcia system not storing a file in a shared folder at a second node until the user at the second node requests the file be downloaded because each of Houston's and Garcia's underlying specifications was designed to be extensible and flexible in their application. *See* Exhibit 1003, 9:40–42 ("the disclosure of the present invention is intended to be illustrative, but not limiting, of the scope of the invention"); Exhibit 1004, 12:27–28 ("the above disclosure of the present invention is intended to be illustrative and not limiting of the invention").

178. Accordingly, the combination of Houston, Garcia, and Manzano renders claims 1–8 and 17–21 obvious and a POSITA would have been motivated to combine the teachings of these references.

C. Grounds 3 and 4: Claims 17–21 are Obvious over Houston, Garcia, and Wu or over Houston, Garcia, Manzano, and Wu

179. As I explained above in Sections IX.A and IX.B, the combination of Houston and Garcia and the combination of Houston, Garcia, and Manzano render obvious claims 17–21, respectively. However, to the extent that the Patent Owner contends that these combinations do not disclose, teach, suggest, or render obvious "*the at least one email . . . is at least partially pre-written*" as recited by element 17[d], Wu teaches this feature.

180. For example, Wu discloses "[a] system and method of producing E-mail [by using templates to] reduce the time consumed for producing E-mail." Exhibit 1006, Abstract; *see also id.*, ¶ 6. Wu explains that "the subject or content of an email may be used repeatedly in some occasions, and it is a waste of time to enter repeatedly the same information." *Id.*, ¶ 5. Accordingly, Wu discloses a system and a method of producing email by using predetermined templates. *See id.*, ¶¶ 7, 8, 17–32, Figures 1 and 2. Wu explains that "common information entered for an email is pre-stored in different databases." *Id.*, ¶ 9. In other words, because Wu's emails

include pre-written content from templates, these emails are at least partially pre-written.

i. Motivation to Combine Houston, Garcia, and Wu or Houston, Garcia, Manzano, and Wu

181. A POSITA would have been motivated to further modify the Houston-Garcia system or the Houston-Garcia system as modified by Manzano (the "Houston-Garcia-Manzano" system) according to and informed by Wu's teachings such that the Houston-Garcia system or the Houston-Garcia-Manzano system would generate an email with body text that is at least partially pre-written.

182. ***First***, implementing the Houston-Garcia system or the Houston-Garcia-Manzano system to generate an email with body text that is at least partially pre-written, as taught by Wu, would have yielded expected, predictable results. The Houston-Garcia system and Houston-Garcia-Manzano system generate an email. *See* Sections IX.A.9.e and IX.B. Wu discloses a system and a method of producing email by using predetermined templates such that the email is partially pre-written. Exhibit 1006, ¶¶ 7–9, 17–32, Figures 1 and 2. A POSITA would have combined the elements of the Houston-Garcia system or the Houston-Garcia-Manzano system with the elements taught by Wu—namely, the email of the Houston-Garcia system or the Houston-Garcia-Manzano system, and the use of templates to generate a partially pre-written email as taught by Wu—using known and routine computer

programming principles to modify the Houston-Garcia system or the Houston-Garcia-Manzano system to generate an email that is at least partially pre-written. This result would have been readily predictable and recognized by a POSITA.

183. ***Second***, the modification of the Houston-Garcia system or the Houston-Garcia-Manzano system to include Wu's use of templates to generate a partially pre-written email would have been readily implemented by simply modifying the Houston-Garcia system or the Houston-Garcia-Manzano system to use of templates to generate a partially pre-written email. A POSITA would have implemented this modification using basic computer programming principles to modify the Houston-Garcia system or the Houston-Garcia-Manzano system to include Wu's teachings of using of templates to generate a partially pre-written email. This modification would have been well within the knowledge and skillset of a POSITA.

184. ***Third***, Wu expressly disclosed the benefits of generating an email with body text that is at least partially pre-written. For instance, Wu's use of templates would have resulted in "producing an E-mail quickly to reduce the time and episodes of entering the related information of an E-mail." Exhibit 1006, ¶ 6. Wu pointed out that "it is a waste of time to enter repeatedly the same information" when "the subject or content of an email may be used repeatedly in some occasions." Exhibit 1006, ¶ 5. As Houston suggests, such occasions would have been present in the Houston-Garcia system or the Houston-Garcia-Manzano system where "any number

of clients 108 may be sharing synchronized folders via host system 110." Exhibit 1003, 3:23–24.

ii. Reasonable Expectation of Success in Combining Houston, Garcia, and Wu or Houston, Garcia, Manzano, and Wu

185. A POSITA would also have had a reasonable expectation of success. The combination of the Houston-Garcia system or the Houston-Garcia-Manzano system with Wu represents a straight-forward implementation of using of templates to generate a partially pre-written body of an email, which a POSITA would have been familiar with and been able to implement. The reasons a POSITA would have expected success parallel those that provide motivation for this combination. *See* Section IX.C.i. Moreover, a POSITA would have had a reasonable expectation of success in implementing the Houston-Garcia system or the Houston-Garcia-Manzano system to use of templates to generate partially pre-written body text of an email because each of Houston's, Garcia's, and Manzano's underlying specification was designed to be extensible and flexible in their application. *See* Exhibit 1003, 9:40–42 ("the disclosure of the present invention is intended to be illustrative, but not limiting, of the scope of the invention"); Exhibit 1004, 12:27–28 ("the above disclosure of the present invention is intended to be illustrative and not limiting of the invention"); Exhibit 1005, ¶ 40 ("the invention is not limited to the details

thereof"; "substitutions and modifications are intended to be embraced within the scope of the invention").

186. Accordingly, the combination of Houston, Garcia, and Wu and the combination of Houston, Garcia, Manzano, and Wu render claims 17–21 obvious and a POSITA would have been motivated to combine the teachings of these references.

X. CONCLUSION

187. Based on the foregoing, it is my opinion that claims 1–8 and 17–21 of the '520 patent are invalid.

XI. AVAILABILITY FOR CROSS EXAMINATION

188. In signing this declaration, I recognize that the declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also recognize that I may be subject to cross examination in the case and that cross examination will take place within the United States. If cross examination is required of me, I will appear for cross examination within the United States during the time allotted for cross examination.

XII. RIGHT TO SUPPLEMENT

189. I reserve the right to supplement my opinions in the future to respond to any arguments that the Patent Owner raises and to take into account new information as it becomes available to me.

XIII. JURAT

190. I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Dated: January 22, 2024

Dr. Todd Mowry

EXHIBIT A

Todd C. Mowry

Curriculum Vitae

December, 2023

Contact Information

Computer Science Department
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

Phone: (412) 268-3725
Fax: (412) 268-5576
Email: tcm@cs.cmu.edu
URL: http://www.cs.cmu.edu/~tcm

Education

Ph.D. in Electrical Engineering, Stanford University, March 1994.

- Thesis: "Tolerating Latency Through Software-Controlled Data Prefetching."
- Supervisors: Anoop Gupta and Monica Lam.

M.S. in Electrical Engineering, Stanford University, June 1989.

B.S. in Electrical Engineering with Highest Distinction, University of Virginia, May 1988.

Academic Appointments

7/08–Present: Professor, Computer Science Department (with a courtesy appointment in the Department of Electrical and Computer Engineering), Carnegie Mellon University.

8/09–6/10: Associate Department Head for Faculty, Computer Science Department

6/15–Present: Director of the Ph.D. program, Computer Science Department

7/97–6/08: Associate Professor, Computer Science Department (with a courtesy appointment in the Department of Electrical and Computer Engineering), Carnegie Mellon University.

12/93–6/97: Assistant Professor, Department of Electrical and Computer Engineering (with a courtesy appointment in the Department of Computer Science), University of Toronto.

9/89–11/93: Graduate Research Assistant, Stanford University.

Honors and Awards

- ASPLOS Influential Paper Award, 2022 (for "Design and Evaluation of a Compiler Algorithm for Prefetching," from ASPLOS 1992).
- Best Paper Runner-Up Award, VLDB conference, 2018.
- ACM Fellow, for contributions to software prefetching and thread-level speculation, 2016.
- SCS Doctoral Dissertation Award co-won by Angela Demke Brown (Ph.D. advisee), 2005.
- Most Thought-Provoking Idea Award (for Claytronics), the Wild and Crazy Idea Session IV held at ASPLOS-XI, 2004.
- Best Paper Award, the 20th International Conference on Data Engineering (ICDE), 2004.
- SCS Doctoral Dissertation Award co-won by J. Gregory Steffan (Ph.D. advisee), 2003.
- Runner-Up for Best Paper Award, the ACM SIGMOD Conference, 2001.
- Alfred P. Sloan Research Fellow, 1999-2001.
- TR100 Award (awarded by MIT's Technology Review magazine to the top 100 most promising young innovators in science and technology), 1999.
- Best Paper Award, the Second Symposium on Operating Systems Design and Implementation, 1996.

Doctoral Thesis Supervision

Student	Thesis Title (or Topic)	Graduation Date
Chi-Keung Luk	Optimizing the Cache Performance of Non-Numeric Applications. *(Nominated for the ACM Thesis Award by the University of Toronto Department of Computer Science.)*	January, 2000
J. Gregory Steffan	Hardware Support for Thread-Level Speculation. *(Co-winner of the SCS Doctoral Dissertation Award.)*	April, 2003
Antonia Zhai	Compiler Optimization of Value Communication for Thread-Level Speculation.	January, 2005
Angela Demke Brown	Explicit Compiler-based Memory Management for Out-of-core Applications. *(Co-winner of the SCS Doctoral Dissertation Award.)*	May, 2005
Christopher Colohan	Applying Thread-Level Speculation to Database Transactions.	November, 2005
Shimin Chen	Redesigning Database Systems in Light of CPU Cache Prefetching.	December, 2005
Amit Manjhi	Increasing the Scalability of Dynamic Web Applications	March, 2008
Olatunji Ruwase	Improving Device Driver Reliability Through Decoupled Dynamic Binary Analysis	May, 2013
Michelle Goodstein	Dataflow Analysis-Based Dynamic Parallel Monitoring	August, 2014
Vivek Seshadri	Simple DRAM and Virtual Memory Abstractions to Enable Highly Efficient Memory Subsystems	May, 2016
Gennady Pekhimenko	Practical Data Compression for Modern Memory Hierarchies	July, 2016
Chris Fallin	Finding and Exploiting Parallelism with Data-Structure-Aware Static and Dynamic Analysis	February, 2019
Prashanth Menon	On Building Robustness into Compilation-Based Main-Memory Database Query Engines	May, 2021
Pratik Fegade	Auto-batching Techniques for Dynamic Deep Learning Computations	January, 2023
Ziqi Wang	Building a More Efficient Cache Hierarchy by Supporting Multiversioning and Leveraging Application-Level Information	January, 2023
Sam Arch	Unifying Declarative and Non-Declarative Code in Relational Database Management Systems	June, 2027 *(expected)*
Hongyi Jin	Compiler Optimizations for Machine Learning Applications	June, 2027 *(expected)*
Ruihang Lai	Compiler Optimizations for Machine Learning Applications	June, 2027 *(expected)*
Patrick Coppock	OS Support for Machine Learning Applications	June, 2027 *(expected)*

Publications

Book Chapters

1. Amir Yazdanbakhsh, Gennady Pekhimenko, Hadi Esmaeilzadeh, Onur Mutlu, and Todd C. Mowry. Towards Breaking the Memory Bandwidth Wall Using Approximate Value Prediction. In *Approximate Circuits: Methodologies and CAD*, Springer International Publishing, 2019, pages 417–441.

Refereed Journal Articles

1. Prashanth Menon, Amadou Ngom, Lin Ma, Todd C. Mowry, and Andrew Pavlo. Permutable Compiled Queries: Dynamically Adapting Compiled Queries without Recompiling. In *Proceedings of the VLDB Endowment*, 14(2):101-113, October 2020.
2. Amir Yazdanbakhsh, Gennady Pekhimenko, Bradley Thwaites, Hadi Esmaeilzadeh, Onur Mutlu, and Todd C. Mowry. RFVP: Rollback-Free Value Prediction with Safe-to-Approximate Loads. In *ACM Transactions on Architecture and Code Optimization (TACO)*, 12(4):62, February 2016.
3. Amir Yazdanbakhsh, Gennady Pekhimenko, Bradley Thwaites, Hadi Esmaeilzadeh, Onur Mutlu, and Todd C. Mowry. Mitigating the Memory Bottleneck with Approximate Load Value Prediction. In *IEEE Design and Test*, 33(1), January 2016.
4. Vivek Seshadri, Samihan Yedkar, Hongyi Xin, Onur Mutlu, Phillip B. Gibbons, Michael A. Kozuch, and Todd C. Mowry. Mitigating Prefetcher-Caused Pollution Using Informed Caching Policies for Prefetched Blocks. In *ACM Transactions on Architecture and Code Optimization (TACO)*, 11(4), January 2015.
5. Shimin Chen, Phillip B. Gibbons, Michael Kozuch, and Todd C. Mowry. Log-Based Architectures: Using Multicore to Help Software Behave Correctly. In *Operating Systems Review*, 45(1), January 2011.
6. Shimin Chen, Michael Kozuch, Phillip B. Gibbons, Michael Ryan, Theodoros Strigkos, Todd C. Mowry, Olatunji Ruwase, Evangelos Vlachos, Babak Falsafi, and Vijaya Ramachandran. Flexible Hardware Acceleration for Instruction-Grain Lifeguards. In *IEEE Micro (Top Picks from 2008 Computer Architecture Conferences)*, 29(1), January 2009.
7. Seth C. Goldstein, Todd C. Mowry, Jason D. Campbell, Michael P. Ashley-Rollman, Michael De Rosa, Stanislav Funiak, James F. Hoburg, Mustafa E. Karagozler, Brian Kirby, Peter Lee, Padmananabhan Pillai, J. Robert Reid, Daniel D. Stancil, and Michael P. Weller. Beyond Audio and Video: Using Claytronics to Enable Pario. In *AI Magazine*, 30(2), March 2009.
8. Antonia Zhai, J. Gregory Steffan, Christopher B. Colohan, and Todd C. Mowry. Compiler and Hardware Support for Reducing the Synchronization of Speculative Threads. In *ACM Transactions on Architecture and Code Optimization (TACO)*, 5(1), May 2008.
9. Christopher B. Colohan, Anastassia Ailamaki, J. Gregory Steffan, and Todd C. Mowry. Incrementally parallelizing database transactions with thread-level speculation. In ACM Transactions on Computer Systems (TOCS), 26(1), February 2008.
10. Shimin Chen, Anastassia Ailamaki, Phillip B. Gibbons, and Todd C. Mowry. Improving Hash Join Performance through Prefetching. In *ACM Transactions on Database Systems*, 32(3):1-32, September 2007.
11. Christopher B. Colohan, Anastasia Ailamaki, J. Gregory Steffan, and Todd C. Mowry. CMP Support for Large and Dependent Speculative Threads. In *IEEE Transactions on Parallel and Distributed Systems*, 18(8):1041-1054, August 2007.
12. J. Gregory Steffan, Christopher B. Colohan, Antonia Zhai and Todd C. Mowry. The STAMPede Approach to Thread-Level Speculation. In *ACM Transactions on Computer Systems*, 23(3):253-300, August 2005.

13. Seth Copen Goldstein, Jason Campbell and Todd C. Mowry. Programmable Matter. In *IEEE Computer*, 38(6):99-101, June 2005.
14. Angela Demke Brown, Todd C. Mowry and Orran Krieger. Compiler-Based I/O Prefetching for Out-of-Core Applications. In *ACM Transactions on Computer Systems*, 19(2):111-170, May 2001.
15. Chi-Keung Luk and Todd C. Mowry. Architectural and Compiler Support for Effective Instruction Prefetching: A Cooperative Approach. In *ACM Transactions on Computer Systems*, 19(1):71-109, February 2001.
16. Todd C. Mowry and Chi-Keung Luk. Understanding Why Correlation Profiling Improves the Predictability of Data Cache Misses in Nonnumeric Applications. In *IEEE Transactions on Computers*, 49(4), April 2000.
17. Chi-Keung Luk and Todd C. Mowry. Automatic Compiler-Inserted Prefetching for Pointer-Based Applications. In *IEEE Transactions on Computers*, 48(2), February 1999.
18. Mark Horowitz, Margaret Martonosi, Todd C. Mowry, and Michael D. Smith. Informing Memory Operations: Memory Performance Feedback Mechanisms and their Applications. In *ACM Transactions on Computer Systems*, 16(2):170-205, May 1998.
19. Todd C. Mowry. Tolerating Latency in Multiprocessors through Compiler-Inserted Prefetching. In *ACM Transactions on Computer Systems*, 16(1):55-92, February 1998.
20. Todd Mowry and Anoop Gupta. Tolerating Latency Through Software-Controlled Prefetching in Shared-Memory Multiprocessors. In *Journal of Parallel and Distributed Computing*, 12(2):87-106, 1991.

Refereed Conference Papers

1. Kai Franz, Samuel Arch, Denis Hirn, Torsten Grust, Todd C. Mowry, and Andrew Pavlo. Dear User-Defined Functions, Inlining Isn't Working Out So Great For Us. Let's Try Batching To Make Our Relationship Work. Sincerely, SQL. To appear in *Proceedings of the 2024 Conference on Innovative Data Systems Research (CIDR)*, January 2024.
2. Siyuan Chen, Pratik Pramod Fegade, Phillip Gibbons, Todd C. Mowry, and Tianqi Chen. ED-Batch: Efficient Automatic Batching of Dynamic Neural Networks via Learned Finite State Machines. In *Proceedings of the Fortieth International Conference on Machine Learning (ICML 2023)*, July 2023.
3. Graham Gobieski, Souradip Ghosh, Marijn Heule, Todd C. Mowry, Tony Nowatzki, Nathan Beckmann, and Brandon Lucia. Riptide: A programmable, energy-minimal dataflow compiler and architecture. In *Proceedings of the 2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, October 2022.
4. Pratik Fegade, Tianqi Chen, Phillip Gibbons, and Todd C. Mowry. The CoRa Tensor Compiler: Compilation for Ragged Tensors with Minimal Padding. In *Proceedings of the Fifth Conference on Machine Learning and Systems (MLSys)*, August 2022.
5. Amadou Ngom, Prashanth Menon, Matthew Butrovich, Lin Ma, Wan Shen Lim, Todd C. Mowry, and Andrew Pavlo. Filter Representation in Vectorized Query Execution. In *Proceedings of the 17th International Workshop on Data Management on New Hardware (DaMoN)*, June 2021.
6. Ziqi Wang, Chul-Hwan Choo, Michael A. Kozuch, Todd C. Mowry, Gennady Pekhimenko, Vivek Seshadri, and Dimitrios Skarlatos. NVOverlay: Enabling Efficient and Scalable High-Frequency Snapshotting to NVM. In *Proceedings of the 48th Annual International Symposium on Computer Architecture (ISCA)*, June 2021.
7. Pratik Fegade, Tianqi Chen, Phillip Gibbons, and Todd C. Mowry. Cortex: A Compiler for Recursive Deep Learning Models. In *Proceedings of the Fourth Conference on Machine Learning and Systems (MLSys)*, April 2021.

8. Daming D. Chen, Phillip B. Gibbons, and Todd C. Mowry. TardisTM: Incremental Repair for Transactional Memory. In *Proceedings of the Eleventh International Workshop on Programming Models and Applications for Multicores and Manycores (PMAM '20)*, February 2020.
9. Ziqi Wang, Michael A. Kozuch, Todd C. Mowry, and Vivek Seshadri. Multiversioned Page Overlays: Enabling Faster Serializable Hardware Transactional Memory. In *Proceedings of the 28th International Conference on Parallel Architectures and Compilation Techniques (PACT)*, September 2019.
10. Vivek Seshadri, Donghyuk Lee, Thomas Mullins, Hasan Hassan, Amirali Boroumand, Jeremie Kim, Michael A. Kozuch, Onur Mutlu, Phillip B. Gibbons, and Todd C. Mowry. Ambit: in-memory accelerator for bulk bitwise operations using commodity DRAM technology. In *Proceedings of the 50th Annual International Symposium on Microarchitecture (MICRO-50)*, October 2017.
11. Prashanth Menon, Andrew Pavlo, and Todd C. Mowry. Relaxed Operator Fusion for In-Memory Databases: Making Compilation, Vectorization, and Prefetching Work Together At Last. In *Proceedings of VLDB 11(1): 1-13*, September 2017.
12. Andrew Pavlo, Gustavo Angulo, Joy Arulraj, Haibin Lin, Jiexi Lin, Lin Ma, Prashanth Menon, Todd Mowry, Matthew Perron, Ian Quah, Siddharth Santurkar, Anthony Tomasic, Skye Toor, Dana Van Aken, Ziqi Wang, Yingjun Wu, Ran Xian, and Tieying Zhang. Self-Driving Database Management Systems" In *Proceedings of the 2017 Conference on Innovative Data Systems Research (CIDR)*, January 2017.
13. Gennady Pekhimenko, Evgeny Bolotin, Nandita Vijaykumar, Onur Mutlu, Todd C. Mowry, and Stephen W. Keckler. A Case for Toggle-Aware Compression for GPU Systems. In *Proceedings of the 22nd International Symposium on High-Performance Computer Architecture (HPCA)*, March 2016.
14. Michelle L. Goodstein, Phillip B. Gibbons, Michael A. Kozuch, and Todd C. Mowry. Tracking and Reducing Uncertainty in Dataflow Analysis-Based Dynamic Parallel Monitoring. In *Proceedings of the 2015 International Conference on Parallel Architectures and Compilation Techniques (PACT '15)*, October 2015.
15. Nandita Vijaykumar, Gennady Pekhimenko, Adwait Jog, Abhishek Bhowmick, Rachata Ausavarungnirun, Chita Das, Mahmut Kandemir, Todd C. Mowry, and Onur Mutlu. A Case for Core-Assisted Bottleneck Acceleration in GPUs: Enabling Efficient Data Compression. In *Proceedings of the 42nd Annual International Symposium on Computer Architecture (ISCA)*, June 2015.
16. Vivek Seshadri, Gennady Pekhimenko, Olatunji Ruwase, Onur Mutlu, Phillip B. Gibbons, Michael A. Kozuch, Todd C. Mowry, and Trishul Chilimbi. Page Overlays: An Enhanced Virtual Memory Framework to Enable Fine-grained Memory Management. In *Proceedings of the 42nd Annual International Symposium on Computer Architecuture (ISCA)*, June 2015.
17. Gennady Pekhimenko, Tyler Huberty, Rui Cai, Onur Mutlu, Phillip P. Gibbons, Michael A. Kozuch, and Todd C. Mowry. Exploiting Compressed Block Size as an Indicator of Future Reuse. In *Proceedings of the 21st International Symposium on High-Performance Computer Architecture (HPCA)*, February 2015.
18. Vivek Seshadri, Abhishek Bhowmick, Onur Mutlu, Phillip B. Gibbons, Michael A. Kozuch, and Todd C. Mowry. The Dirty-Block Index. In *Proceedings of the 41st Annual International Symposium on Computer Architecuture (ISCA)*, June 2014.
19. Olatunji Ruwase, Michael A. Kozuch, Phillip B. Gibbons, and Todd C. Mowry. Guardrail: High Fidelity Correctness Checking of Device Drivers for Safeguarding I/O Operations. In *Proceedings of the Nineteenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2014)*, March 2014.

20. Gennady Pekhimenko, Vivek Seshadri, Yoongu Kim, Hongyi Xin, Onur Mutlu, Phillip B. Gibbons, Michael A. Kozuch, and Todd C. Mowry. Linearly Compressed Pages: A Low-Complexity, Low-Latency Main Memory Compression Framework. In *Proceedings of the 46th Annual International Symposium on Microarchitecture (MICRO-46)*, December 2013.

21. Vivek Seshadri, Yoongu Kim, Chris Fallin, Donghyuk Lee, Rachata Ausavarungnirun, Gennady Pekhimenko, Yixin Luo, Onur Mutlu, Phillip B. Gibbons, Michael A. Kozuch, and Todd C. Mowry. RowClone: Fast and Energy-Efficient In-DRAM Bulk Data Copy and Initialization. In *Proceedings of the 46th Annual International Symposium on Microarchitecture (MICRO-46)*, December 2013.

22. Michelle L. Goodstein, Shimin Chen, Phillip B. Gibbons, Michael A. Kozuch, and Todd C. Mowry. Chrysalis Analysis: Incorporating Synchronization Arcs in Dataflow-Analysis-based Parallel Monitoring. In *Proceedings of the 21st International Conference on Parallel Architectures and Compilation Techniques (PACT-2012)*, September 2012.

23. Vivek Seshadri, Onur Mutlu, Todd C. Mowry, and Michael A. Kozuch. The Evicted-Address Filter: A Unified Mechanism to Address Both Cache Pollution and Thrashing. In *Proceedings of the 21st International Conference on Parallel Architectures and Compilation Techniques (PACT-2012)*, September 2012.

24. Gennady Pekhimenko, Vivek Seshadri, Onur Mutlu, Todd C. Mowry, Phillip B. Gibbons, and Michael A. Kozuch. Base-Delta-Immediate Compression: A Practical Data Compression Mechanism for On-Chip Caches. In *Proceedings of the 21st International Conference on Parallel Architectures and Compilation Techniques (PACT-2012)*, September 2012.

25. Olatunji Ruwase, Shimin Chen, Phillip Gibbons, and Todd C. Mowry. Decoupled Lifeguards: Enabling Path Optimizations for Dynamic Correctness Checking Tools. In *Proceedings of the ACM SIGPLAN 2010 Conference on Programming Language Design and Implementation (PLDI)*, June 2010.

26. F. Ryan Johnson, Radu Stoica, Anastasia Ailamaki, and Todd C. Mowry. Decoupling Contention Management from Scheduling. In *Proceedings of the Fifteenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2010)*, March 2010.

27. Michelle Goodstein, Evangelos Vlachos, Shimin Chen, Phillip B. Gibbons, Michael Kozuch, and Todd C. Mowry. Butterfly Analysis: Adapting Dataflow Analysis to Dynamic Parallel Monitoring. In *Proceedings of the Fifteenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2010)*, March 2010.

28. Evangelos Vlachos, Michelle Goodstein, Michael Kozuch, Shimin Chen, Babak Falsafi, Phillip B. Gibbons, and Todd C. Mowry. ParaLog: Enabling and Accelerating Online Parallel Monitoring of Multithreaded Applications. In *Proceedings of the Fifteenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2010)*, March 2010.

29. Amit Manjhi, Charles Garrod, Bruce M. Maggs, Todd C. Mowry, Anthony Tomasic. Holistic Query Transformations for Dynamic Web Applications. In *Proceedings of the 2009 IEEE 25th International Conference on Data Engineering (ICDE)*, March-April 2009.

30. Daniel J. Dewey, Michael P. Ashley-Rollman, Michael DeRosa, Seth Copen Goldstein, Todd C. Mowry, Siddhartha S. Srinivasa, Padmanabhan Pillai, and Jason Campbell. Generalizing metamodules to simplify planning in modular robotic systems. In *Proceedings of the IEEE/RSJ 2008 International Conference on Intelligent Robots and Systems (IROS)*, September 2008.

31. Charles Garrod, Amit Manjhi, Anastasia Ailamaki, Bruce Maggs, Todd Mowry, Christopher Olston, and Anthony Tomasic. Scalable Query Result Caching for Web Applications. In *Proceedings of the 34th International Conference on Very Large Databases (VLDB)*, August 2008.

32. Lei Li, Wenjie Fu, Fan Guo, Todd C. Mowry, and Christos Faloutsos. Cut-and-stitch: efficient parallel learning of linear dynamical systems on SMPs. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, August 2008.

33. Shimin Chen, Michael Kozuch, Theodoros Strigkos, Babak Falsafi, Phillip B. Gibbons, Todd C. Mowry, Michael Ryan, Olatunji Ruwase, and Evangelos Vlachos. Flexible Hardware Acceleration for Instruction-Grain Program Monitoring. In *Proceedings of the 35th Annual International Symposium on Computer Architecture (ISCA)*, June 2008.

34. Olatunji Ruwase, Phillip B. Gibbons, Todd C. Mowry, Vijaya Ramachandran, Shimin Chen, Michael Kozuch, and Michael Ryan. Parallelizing dynamic information flow tracking. In *Proceedings of the 20th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, June 2008.

35. Brian Kirby, Burak Aksak, James Hoburg, Todd C. Mowry, and Padmanabhan Pillai. A Modular Robotic System Using Magnetic Force Effectors. In *Proceedings of the IEEE/RSJ 2007 International Conference on Intelligent Robots and Systems (IROS)*, October 2007.

36. Michael Ashley-Rollman, Seth Goldstein, Peter Lee, Todd C. Mowry, and Padmanabhan Pillai. Meld: A Declarative Approach to Programming Ensembles. In *Proceedings of the IEEE/RSJ 2007 International Conference on Intelligent Robots and Systems (IROS)*, October 2007.

37. Amit Manjhi, Phillip B. Gibbons, Anastassia Ailamaki, Charles Garrod, Bruce M. Maggs, Todd C. Mowry, Christopher Olston, Anthony Tomasic, and Haifeng Yu. Invalidation Clues for Database Scalability Services. In *Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering (ICDE)*, pages 316-325, April 2007.

38. Michael De Rosa, Peter Lee, Seth Goldstein, Jason Campbell, Padmanabhan Pillai, and Todd C. Mowry. Distributed Watchpoints: Debugging Large Multi-Robot Systems. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3723-3729, April 2007.

39. Benjamin D. Rister, Jason Campbell, Padmanabhan Pillai, and Todd C. Mowry. Integrated Debugging of Large Modular Robot Ensembles. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2227-2234, April 2007.

40. Shimin Chen, Phillip Gibbons, Michael Kozuch, Vasileios Liaskovitis, Anastassia Ailamaki, Guy Blelloch, Babak Falsafi, Limor Fix, Nikos Hardavellas, Todd C. Mowry and Chris Wilkerson. Scheduling Threads for Constructive Cache Sharing on CMPs. In *Proceedings of 18th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, pages 105-115, June 2007.

41. Shimin Chen, Babak Falsafi, Phillip B. Gibbons, Michael Kozuch, Todd C. Mowry, Radu Teodorescu, Anastassia Ailamaki, Limor Fix, Gregory R. Ganger, Bin Lin, and Steven W. Schlosser. Log-Based Architectures for General-Purpose Monitoring of Deployed Code. In *Proceedings of the Workshop on Architectural and System Support for Improving Software Dependability (ASID), held with ASPLOS XII*, October 2006.

42. Christopher B. Colohan, Anastassia Ailamaki, J. Gregory Steffan, and Todd C. Mowry. Tolerating Dependences Between Large Speculative Threads Via Sub-Threads. In *Proceedings of the 33rd Annual International Symposium on Computer Architecture (ISCA)*, pages 216-226, June 2006.

43. Amit Manjhi, Anastassia Ailamaki, Bruce M. Maggs, Todd C. Mowry, Christopher Olston, and Anthony Tomasic. Simultaneous Scalability and Security for Data-Intensive Web Applications. In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*, pages 241-252, June 2006.

44. Vasileios Liaskovitis, Shimin Chen, Phillip B. Gibbons, Anastassia Ailamaki, Guy E. Blelloch, Babak Falsafi, Limor Fix, Nikos Hardavellas, Michael Kozuch, Todd C. Mowry, and Chris Wilkerson. Parallel Depth First vs. Work Stealing Schedulers on CMP Architectures. In

Proceedings of 18th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), August 2006.

45. Brian Kirby, Jason Campbell, Burak Aksak, Padmanabhan Pillai, James F. Hoburg, Todd C. Mowry, and Seth Copen Goldstein. Catoms: Moving Robots without Moving Parts. In *Proceedings of the Twentieth National Conference qon Artificial Intelligence (AAAI)*, pages 1730-1731, July 2005.
46. Christopher Olston, Amit Manjhi, Charles Garrod, Anastassia Ailamaki, Bruce M. Maggs, and Todd C. Mowry. A Scalability Service for Dynamic Web Applications. In *Proceedings of the Second Biennial Conference on Innovative Data Systems Research (CIDR)*, pages 56-69, January 2005.
47. Christopher B. Colohan, Anastassia Ailamaki, J. Gregory Steffan, and Todd C. Mowry. Optimistic Intra-Transaction Parallelism on Chip Multiprocessors. In *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB)*, pages 73-84, September 2005.
48. Shimin Chen, Anastassia Ailamaki, Phillip B. Gibbons, and Todd C. Mowry. Inspector Joins. In *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB)*, pages 817-828, September 2005.
49. Antonia Zhai, Christopher B. Colohan, J. Gregory Steffan, and Todd C. Mowry. Compiler Optimization of Memory-Resident Value Communication Between Speculative Threads. In *Proceedings of the 2007 International Symposium on Code Generation and Optimization (CGO)*, pages 39-52, March 2004.
50. Shimin Chen, Anastassia Ailamaki, Phillip B. Gibbons and Todd C. Mowry. Improving Hash Join Performance through Prefetching. In *Proceedings of the 20th International Conference on Data Engineering (ICDE)*, pages 116-127, April 2004. *(This paper received the Best Paper Award.)*
51. Antonia Zhai, Christopher B. Colohan, J. Gregory Steffan and Todd C. Mowry. Compiler Optimization of Scalar Value Communication Between Speculative Threads. In *Proceedings of the Tenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X)*, October 2002.
52. Shimin Chen, Philip B. Gibbons, Todd C. Mowry and Gary Valentin. Fractal Prefetching B+-Trees: Optimizing Both Cache and Disk Performance. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, June 2002.
53. J. Gregory Steffan, Christopher B. Colohan, Antonia Zhai and Todd C. Mowry. Improving Value Communication for Thread-Level Speculation. In *Proceedings of the Eighth International Symposium on High-Performance Computer Architecture (HPCA)*, February 2002.
54. Shimin Chen, Philip B. Gibbons and Todd C. Mowry. Improving Index Performance through Prefetching. In *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, pages 235-246, May 2001. *(This paper received the Runner-Up for Best Paper Award.)*
55. Angela Demke Brown and Todd C. Mowry. Taming the Memory Hogs: Using Compiler-Inserted Releases to Manage Physical Memory Intelligently. In *Proceedings of the Fourth Symposium on Operating Systems Design and Implementation (OSDI 2000)*, pages 31-44, October 2000.
56. J. Gregory Steffan, Christopher B. Colohan, Antonia Zhai and Todd C. Mowry. A Scalable Approach to Thread- Level Speculation. In *Proceedings of the 27th Annual International Symposium on Computer Architecture*, pages 1-12, June 2000.
57. Todd C. Mowry and Sherwyn R. Ramkissoon. Software-Controlled Multithreading Using Informing Memory Operations. In *Proceedings of the Sixth International Symposium on High-Performance Computer Architecture*, January, 2000.

58. Chi-Keung Luk and Todd C. Mowry. Memory Forwarding: Enabling Aggressive Layout Optimizations by Guaranteeing the Safety of Data Relocation. In *Proceedings of the 26th Annual International Symposium on Computer Architecture (ISCA)*, pages 88-99, May 1999.
59. Chi-Keung Luk and Todd C. Mowry. Cooperative Prefetching: Compiler and Hardware Support for Effective Instruction Prefetching in Modern Microprocessors. In *Proceedings of the 31st Annual International Symposium on Microarchitecture*, pages 182-193, December 1998.
60. J. Gregory Steffan and Todd C. Mowry. The Potential for Using Thread-Level Data Speculation to Facilitate Automatic Parallelization. In *Proceedings of the Fourth International Symposium on High-Performance Computer Architecture*, pages 2-13, February, 1998.
61. Charles Chan, Adley Lo, and Todd C. Mowry. Comparative Evaluation of Latency Tolerance Techniques for Software Distributed Shared Memory. In *Proceedings of the Fourth International Symposium on High-Performance Computer Architecture*, pages 300-311, February, 1998.
62. Todd C. Mowry and Chi-Keung Luk. Predicting Data Cache Misses in Non-Numeric Applications Through Correlation Profiling. In *Proceedings of the 30th Annual International Symposium on Microarchitecture*, pages 314-320, December 1997.
63. Todd C. Mowry, Angela K. Demke and Orran Krieger. Automatic Compiler-Inserted I/O Prefetching for Out-of-Core Applications. In *Proceedings of the Second Symposium on Operating Systems Design and Implementation (OSDI '96)*, pages 3-17, October 1996. *(This paper received the Best Paper Award.)*
64. Chi-Keung Luk and Todd C. Mowry. Compiler-Based Prefetching for Recursive Data Structures. In *Proceedings of the Seventh International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 222-233, October 1996.
65. Edouard Bugnion, Jennifer M. Anderson, Todd C. Mowry, Mendel Rosenblum and Monica S. Lam. Compiler- Directed Page Coloring for Multiprocessors. In *Proceedings of the Seventh International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 244-255, October 1996.
66. Mark Horowitz, Margaret Martonosi, Todd C. Mowry, and Michael D. Smith. Informing Memory Operations: Providing Memory Performance Feedback in Modern Processors. In *Proceedings of the 23rd Annual International Symposium on Computer Architecture*, pages 260-270, May 1996.
67. Todd C. Mowry, Monica S. Lam and Anoop Gupta. Design and Evaluation of a Compiler Algorithm for Prefetching. In *Proceedings of the Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 62-73, October 1992.
68. Anoop Gupta, John Hennessy, Kourosh Gharachorloo, Todd Mowry, and Wolf-Dietrich Weber. Comparative Evaluation of Latency Reducing and Tolerating Techniques. In *Proceedings of the 18th Annual International Symposium on Computer Architecture*, pages 254-263, May 1991.
69. Anoop Gupta, Wolf-Dietrich Weber, and Todd Mowry. Reducing Memory and Traffic Requirements for Scalable Directory-Based Cache Coherence Schemes. In *Proceedings of International Conference on Parallel Processing*, pages 312-321, August 1990.

Technical Reports

- Evangelos Vlachos, Michelle Goodstein, Michael Kozuch, Shimin Chen, Babak Falsafi, Phillip B. Gibbons, Todd C. Mowry, and Olatunji Ruwase. Parallel LBA: Conherence-based Parallel Monitoring of Multithreaded Applications. Carnegie Mellon University Technical Report CMU-CS-09-108, March 2009.
- Michelle Goodstein, Evangelos Vlachos, Shimin Chen, Phillip Gibbons, Michael Kozuch, and Todd C. Mowry. The Butterfly Model: Theoretical Foundations. Carnegie Mellon University Technical Report CMU-CS-08-170, February 2009.

- Amit Manjhi, Anastassia Ailamaki, Bruce M. Maggs, Todd C. Mowry, Christopher Olston, and Anthony Tomasic. Simultaneous Scalability and Security for Data-Intensive Web Applications. Carnegie Mellon University Technical Report CMU-CS-06-116, March 2006.
- Charles Garrod, Amit Manjhi, Anastassia Ailamaki, Phillip Gibbons, Bruce Maggs, Todd Mowry, Christopher Olston, and Anthony Tomasic. Scalable Consistency Management for Web Database Caches. Carnegie Mellon University Technical Report CMU-CS-06-128, July 2006.
- Amit Manjhi, Phillip B. Gibbons, Anastassia Ailamaki, Charles Garrod, Bruce M. Maggs, Todd C. Mowry, Christopher Olston, Anthony Tomasic, and Haifeng Yu. Invalidation Clues for Database Scalability Services. Carnegie Mellon University Technical Report CMU-CS-06-139, July 2006.
- Christopher B. Colohan, Anastassia Ailamaki, J. Gregory Steffan, and Todd C. Mowry. Supporting Large Speculative Threads for Databases and Beyond. Carnegie Mellon University Technical Report CMU-CS-05-109, July 2005.
- Christopher B. Colohan, Anastassia Ailamaki, J. Gregory Steffan, and Todd C. Mowry. Optimistic Intra-Transaction Parallelism on Chip Multiprocessors. Carnegie Mellon University Technical Report CMU-CS-05-118, March 2005.
- Shimin Chen, Anastassia Ailamaki, Phillip B. Gibbons, and Todd C. Mowry. Improving Hash Join Performance through Prefetching. Carnegie Mellon University Technical Report CMU-CS-03-157, October 2003.
- Shimin Chen, Phillip B. Gibbons, Todd C. Mowry, and Gary Valentin. Fractal Prefetching B+-Trees: Optimizing Both Cache and Disk Performance. Carnegie Mellon University Technical Report CMU-CS-02-115, March 2002.
- Spiros Papadimitrious and Todd C. Mowry. Exploring Thread-Level Speculation in Software: The Effects of Memory Access Tracking Granularity. Carnegie Mellon University Technical Report CMU-CS-01-145, July 2001.
- Shimin Chen, Phillip B. Gibbons and Todd C. Mowry. Improving Index Performance through Prefetching. Carnegie Mellon University Technical Report CMU-CS-00-177, December 2000.
- J. Gregory Steffan, Christopher B. Colohan and Todd C. Mowry. Extending Cache Coherence to Support Thread-Level Data Speculation on a Single Chip and Beyond. Carnegie Mellon University Technical Report CMU-CS-98-171, December 1998.
- Todd C. Mowry and Sherwyn R. Ramkissoon. Software-Controlled Multithreading Using Informing Memory Operations. Carnegie Mellon University Technical Report CMU-CS-98-169, October 1998.
- Chi-Keung Luk and Todd C. Mowry. Compiler and Hardware Support for Automatic Instruction Prefetching: A Cooperative Approach. Carnegie Mellon University Technical Report CMU-CS-98-140, June 1998.
- J. Gregory Steffan, Christopher B. Colohan and Todd C. Mowry. Architectural Support for Thread-Level Data Speculation. Carnegie Mellon University Technical Report CMU-CS-97-188, November 1997.
- Todd C. Mowry and Chi-Keung Luk. Predicting Data Cache Misses in Non-Numeric Applications Through Correlation Profiling. Carnegie Mellon University Technical Report CMU-CS-97-175, September 1997.
- J. Gregory Steffan and Todd C. Mowry. The Potential for Thread-Level Data Speculation in Tightly-Coupled Multiprocessors. University of Toronto Technical Report CSRI-TR-350, February 1997.
- Chi-Keung Luk and Todd C. Mowry. Predicting Data Cache Misses in Non-Numeric Applications Through Correlation Profiling. University of Toronto Technical Report CSRI-TR-359, February, 1997.

- Mark Horowitz, Margaret Martonosi, Todd C. Mowry, and Michael D. Smith. Informing Loads: Enabling Software to Observe and React to Memory Behavior. Stanford University Technical Report CSL-TR-95-673, July 1995.
- Todd C. Mowry. Tolerating Latency Through Software-Controlled Data Prefetching. Technical Report CSL-TR-94-628, Stanford University, June 1994.

Patents Held

- Todd C. Mowry. U.S. Patent 7,127,586: Prefetching hints. Issued October, 2006.
- Shimin Chen, Phillip B. Gibbons, and Todd C. Mowry. U.S. Patent 6,772,179: System and method for improving index performance through prefetching. Issued August, 2004.
- Todd C. Mowry. U.S. Patent 6,240,488: Prefetching hints. Issued May, 2001.
- Todd C. Mowry. U.S. Patent 5,732,242: Consistently specifying way destinations through prefetching hints. Issued March, 1998.
- Todd C. Mowry and Earl A. Killian. U.S. Patent 5,696,958: Method and apparatus for reducing delays following the execution of a branch instruction in an instruction pipeline. Issued December, 1997.

Distinguished Lectures

9/25/13: "The Log-Based Architectures Project: Exploiting Multicore to Help Parallel Software Behave Correctly", University of Wisconsin-Madison, Department of Computer Science, Distinguished Lecture Series.

2/7/08: "Pario: the Next Step Beyond Audio and Video", University of Toronto, Department of Electrical and Computer Engineering Distinguished Lecture Series.

Professional Activities

Professional Societies

- Member of the Institute of Electrical and Electronics Engineers (IEEE)
- Member of the Association of Computing Machinery (ACM)

Industrial Employment

5/04–6/07: Intel Corporation, Pittsburgh, Pennsylvania: Director of the Intel Research Pittsburgh Lab.

6/89–11/93: Silicon Graphics, Inc., Mountain View, California (formerly MIPS Computer Systems, Sunnyvale California): Computer Architect (part-time).

Consulting

7/07–2/11: Intel Corporation, Pittsburgh, Pennsylvania: Research Advisor.

8/96–4/04: SandCraft, Inc., Santa Clara, California: Member of the Technical Advisory Board.

1/96–4/04: IBM, Toronto: Visiting Scientist.

12/93–12/96: Silicon Graphics, Inc., Mountain View, California: Computer Architecture Consultant.

University Committee Work and Other Service Activities

Committee Work

- Faculty Hiring Committees:
  - Computer Science Department: 2008, 2010.
  - Computer Systems Area: 2000, 2001, 2002, 2003, 2004.
- Chair of the Computer Science Department Head Selection Committee, 2007.
- CSD Ph.D. Admissions Committee, 2001, 2002, 2003.
- School of Computer Science Council, 2001-2004.
- Carnegie Mellon University Faculty Senate (Presidential Appointeee), 2000-2001.
- Computer Systems Area Advocate, 2000-2004.
- Doctoral Review Committee, 1998-2004.
- ACM Thesis Award Nomination Committee, 1999.

Journal, Conference and Workshop Organization

- Editor-in-Chief, ACM Transactions on Computer Systems, 2013-2018.
- Associate Editor, ACM Transactions on Computer Systems, 2001-2013.
- Program Chair of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), 2011.
- Sponsorship Chair, International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), 2010.
- Member of the ISCA program committee, 1998, 2000, and 2011.
- Member of the ASPLOS program committee, 2000, 2004.
- Member of the ASPLOS external review committee, 2012.
- Co-Program Chair of the International Conference on Parallel Architectures and Compilation Techniques (PACT), 2001.
- Member of the International Symposuim on Microarchitecture program committee, 1998.
- Member of the Workshop on Architectural and System Support for Improving Software Dependability (ASID) program committee, 2005.
- Member of the Workshop on Memory System Performance (MSP) program committee (in conjunction with PLDI), 2002 and 2004.
- Member of the IBM CASCON program committee, 1995, 1996, 1997.
- Member of the First SUIF Workshop Program Committee, 1996.
- Organized the CASCON 97 workshop on Compiler Optimization, 1997.
- Initiated and organized the 1st Toronto / Rochester Workshop on Shared-Memory Multiprocessors, 1994.