

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

DROPBOX, INC.,
Petitioner,

v.

ENTANGLED MEDIA, LLC
Patent Owner.

Case No. IPR2024-00284
U.S. Patent No. 8,296,338

DECLARATION OF PATRICK D. MCDANIEL, PH.D.

U.S. PATENT NO. 8,296,338 (CLAIMS 1-14)

Contents

I.	ENGAGEMENT	1
II.	PROFESSIONAL BACKGROUND.....	1
III.	MATERIALS REVIEWED	6
IV.	DESCRIPTION OF THE RELEVANT TIMEFRAME, THE RELEVANT FIELD, AND A PERSON OF ORDINARY SKILL IN THE ART.....	6
V.	TECHNOLOGY BACKGROUND.....	7
A.	What are “File Systems”?	7
B.	The Evolution of File Systems.....	11
1.	From Files to Folders	11
2.	Remote Storage for File Systems.....	12
3.	Sharing Files with Peers.....	14
C.	The Fundamentals of File Systems	16
1.	The File System API.....	17
2.	Modifying File Systems	20
D.	Distributed File Systems in Peer-to-Peer Networks	22
VI.	OVERVIEW OF THE ’338 PATENT	24
VII.	OVERVIEW OF THE PRIOR ART	27
A.	Aboulhosn	27
B.	Phillips.....	29
C.	Shappell	29
D.	Tsang	30
VIII.	CLAIM CONSTRUCTION	32
A.	Terms to Be Construed.....	32
1.	“web-based system”	32
IX.	ANTICIPATION	33
X.	OBVIOUSNESS.....	34
XI.	OPINIONS REGARDING PATENTABILITY	36

XII.	MY OPINIONS AS TO THE CHALLENGED CLAIMS OF THE	
	'338 PATENT.....	38
A.	My Opinions as to Claims 1-2 and 5-14 in view of the	
	Aboulhosn-Phillips combination.....	38
1.	Claim 1	38
a.	1[P].....	39
b.	1[A].....	42
c.	1[B]	51
d.	1[C]	55
e.	1[D].....	58
f.	1[E]	64
g.	1[F].....	66
h.	1[G].....	69
i.	1[H].....	75
j.	1[I]	85
k.	1[J]	87
2.	Motivation to Combine Aboulhosn and Phillips	92
3.	Claim 2	103
4.	Claim 5	105
5.	Claim 6	106
6.	Claim 7	107
a.	7[P].....	108
b.	7[A].....	109
c.	7[B]	110
d.	7[C]	111
e.	7[D].....	111
f.	7[E]	112
g.	7[F].....	113
h.	7[G].....	114

7.	Claim 8.....	115
8.	Claim 9.....	115
9.	Claim 10.....	116
	a. 10[P].....	117
	b. 10[A-E].....	118
10.	Claim 11.....	119
11.	Claim 12.....	120
12.	Claim 13.....	121
13.	Claim 14.....	122
B.	My Opinions as to Claims 3 and 4 based on Aboulhosn-Phillips in view of Tsang.....	122
	1. Claim 3.....	122
	2. Claim 4.....	127
	3. Motivation to Combine Aboulhosn-Phillips with Tsang.....	128
C.	My Opinions as to Claims 1-2 and 5-14 in view of the Aboulhosn-Phillips-Shappell combination	133
	1. Claims 1-2, 5, and 7-14.....	133
	2. Motivation to Combine Aboulhosn-Phillips with Shappell.....	135
	3. Claim 6.....	137
D.	My Opinions as to Claims 3 and 4 in view of the Aboulhosn- Phillips-Tsang in view of Shappell	138
XIII.	CONCLUDING STATEMENTS.....	139

EXHIBIT LIST

EXHIBITS FILED BY PETITIONERS	
Ex-1001	U.S. Patent No. 8,296,338 (the “’338 patent”)
Ex-1002	File History of the ’338 patent
Ex-1003	Declaration of Dr. Patrick McDaniel
Ex-1004	CV of Dr. Patrick McDaniel
Ex-1005	U.S. Patent No. 6,938,042 (“Aboulhoshn”)
Ex-1006	U.S. Patent No. 7,058,696 (“Phillips”)
Ex-1007	U.S. Patent Publication No. 2005/0091595 A1 (“Shappell”)
Ex-1008	U.S. Patent No. 7,930,270 (“Tsang”)
Ex-1009	U.S. Patent Application No. 10/115,448 (“448 Application”)
Ex-1010	U.S. Patent No. 8,484,260 (the “’260 patent”)
Ex-1011	File History of the ’260 patent
Ex-1012	Catherine Soanes and Angus Stevenson, Concise Oxford English Dictionary, Eleventh Edition, Oxford University Press, 2004

Ex-1013	Steven M. Kaplan, Wiley Electrical and Electronics Engineering Dictionary, John Wiley & Sons, 2004.
Ex-1014	RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1
Ex-1015	RFC 2818 – HTTP Over TLS
Ex-1016	Webpage printout – Rus Shuler, “How Does the Internet Work?” (2002)
Ex-1017	Wayback printout – “What do you need to know about how the Internet Works”
Ex-1018	Wayback printout – “File system”, Wikipedia
Ex-1019	U.S. Patent No. 9,400,884
Ex-1020	Merriam-Webster’s Collegiate Dictionary, Eleventh Edition, 2003
Ex-1021	Wayback printout – “Central processing unit”, Wikipedia
Ex-1022	U.S. Patent No. 6,351,776
Ex-1023	Good, Nathaniel S., and Aaron Krekelberg. “Usability and privacy: a study of Kazaa P2P file-sharing.” Proceedings of the SIGCHI conference on Human factors in computing systems. 2003.
Ex-1024	McCarty, William. “Learning Debian GNU/Linux.” (1999), Chapter 10 Setting Up a Linux-Based LAN, Section 10.3 Samba

Ex-1025	McCarty, William. “Learning Debian GNU/Linux.” (1999), Chapter 12 Setting Up a Linux-Based WAN, Section 12.2 Installing and Configuring a Web Server
Ex-1026	Wayback printout – “World Wide Web”, Wikipedia
Ex-1027	Wayback printout – “Hypertext Transfer Protocol”, Wikipedia
Ex-1028	Wayback printout – M.S. Schmalz, “Organization of Computer Systems: Introduction, Abstractions, Technology”
Ex-1029	Wayback printout – Chester F. Carlson, The Center for Imaging Science, Basics of File Structure
Ex-1030	Nagar, Rajeev. Windows NT file system internals: a developer's guide. O'Reilly & Associates, Inc., 1997. (excerpted)
Ex-1031	Wayback printout – “Filesystems”
Ex-1032	Webpage printout – “Multics”
Ex-1033	Sandberg, Russel, et al. “Design and implementation of the Sun network filesystem.” Proceedings of the summer 1985 USENIX conference. 1985.
Ex-1034	[Reserved]
Ex-1035	Wayback printout – Napster
Ex-1036	Webpage printout - Peer-to-peer interactions in web brokering systems

Ex-1037	Wayback printout – Introduction to Linux Loadable Kernel Modules
Ex-1038	Webpage printout – Deploying your hardware and software systems
Ex-1039	Dolstra, Eelco, Eelco Visser, and Merijn De Jonge. “Imposing a memory management discipline on software deployment.” Proceedings. 26th International Conference on Software Engineering. IEEE, 2004.
Ex-1040	U.S. Patent Publication No. 2007/0283050 A1 (“Savage”)
Ex-1041	Foldershare Guide, available at: https://web.archive.org/web/20060313171107/https://www.foldershare.com/download/files/FolderShareGuide.pdf
Ex-1042	U.S. Patent No. 9,032,295 (“Patel”)
Ex-1043	Sotera Stipulation
Ex-1044	Wayback printout – net2ftp
Ex-1045	Wayback printout - TXFTP
Ex-1046	Wayback printout – Google Accounts

I, Patrick D. McDaniel, Ph.D., declare as follows:

I. ENGAGEMENT

1. I have been retained by Dropbox, Inc. (“Dropbox” or “Petitioner”) in connection with the above-captioned petition for *Inter Partes* Review (“IPR”) of U.S. Patent No. 8,296,338 to Erik Caso and Michael Abraham (“the ’338 Patent,” Ex. 1001). I understand the ’338 Patent is currently assigned to Entangled Media, LLC (“Patent Owner”).

2. I have been asked by Petitioner to offer opinions regarding the ’338 Patent, including the unpatentability of claims 1-14 (which I may refer to subsequently as the “challenged claims”) in view of certain prior art. This declaration sets forth the opinions I have reached to date regarding these matters.

3. I am being compensated by Petitioner at my standard hourly consulting rate for my time spent on this matter. My compensation is not contingent on the outcome of the IPR or on the substance of my opinions.

4. I have no financial interest in Petitioner or Patent Owner.

II. PROFESSIONAL BACKGROUND

5. My qualifications are set forth in my curriculum vitae, a copy of which is attached as an Exhibit 1004. Exhibit 1004 also includes a list of my publications and the cases in which I have testified at deposition, hearing, or trial within the past four years. As set forth in my curriculum vitae:

6. I earned my Ph.D. in Computer Science from the Computer Science and Engineering Department at University of Michigan, Ann Arbor in 2001. I earned my Bachelor of Science degree in Computer Science from Ohio University in 1989, and my Master of Science degree, also in Computer Science, from Ball State University in 1991.

7. Since 2022, I have been Tsun-Ming Shih Professor of Computer Sciences in the School of Computer, Data and Information Sciences at the University of Wisconsin-Madison. Prior to my move to Wisconsin, I was the William L. Weiss Professor of Information and Communications Technology in the School of Electrical Engineering and Computer Science at the Pennsylvania State University in University Park, PA. I was also the director of the Institute for Network and Security Research, and founder and co-director of the Systems and Internet Infrastructure Security Laboratory, a research laboratory focused on the study of security in diverse network and computer environments. My research efforts primarily involve computer systems, network management and authentication, systems security, and technical public policy.

8. I frequently teach and do research on filesystems and storage, particularly as it relates to security. For example, Chapter 6 of my PhD dissertation, “Case Studies: Virtual Private Filesystems in AMirD”, described a peer-to-peer distributed file system that allowed multiple devices to synchronize

and share metadata indices and file data blocks amongst devices. I continue to do research in filesystems to this day, and in my most recent work (appearing in the technical report Securing Cloud File Systems using Shielded Execution) has developed techniques for securing filesystem metadata on untrusted hardware. I also frequently teach filesystems design in my systems programming course.

9. Prior to 2017, I was an Assistant Professor (2004-2007), Associate Professor (2007-2011), Full Professor (2011-2015), and Distinguished Professor (2016-2017) of Computer Science and Engineering and the William L. Weiss Professor of Information and Communications Technology (2017-2022) at the Pennsylvania State University. Since 2004, I have taught courses in the field of computer systems, systems programming, mobile device security, networks, and network and computer security at both the undergraduate and graduate level. I created and maintained several of these courses at Penn State University. I have begun teaching courses at the University of Wisconsin-Madison in the Fall of 2023, including courses such as network and systems security.

10. From 2003-2009, I was an Adjunct Professor at the Stern School of Business at New York University in New York, NY. At the Stern School of Business, I taught courses in computer and network security and online privacy.

11. I am a Fellow of the Association for Computing Machinery (ACM, the leading professional association for computer science), a Fellow Institute for

Electrical and Electronics Engineering (IEEE, the leading professional association for engineering), and Fellow of the American Association for the Advancement of Science (AAAS, the leading professional association for science).

12. I am the director of the NSF Frontier Center for Trustworthy Machine Learning (CTML), a research organization exploring security and privacy of machine learning (and AI in general). Funded by the US National Science Foundation, the CTML is led by the University of Wisconsin-Madison and has members from Stanford, the University of California, Berkeley, The University of California-San Diego, and the University of Virginia.

13. I was the Program Manager (PM) and lead scientist for the Cyber Security Collaborative Research Alliance (CRA). The CRA is led by Penn State University and includes faculty and researchers the Army Research Laboratory, Carnegie Mellon University, Indiana University, the University of California-Davis, and the University of California-Riverside. This initiative is a major research project aimed at developing a new science of cyber-security for military networks, computers, and installations.

14. I have served as an advisor to several Ph.D. and master's degree candidates, several of whom have gone on to become professors at various institutions such as North Carolina State University, Purdue University, the University of Toronto, the University of Oregon, and the Georgia Institute of

Technology and the University of Florida. I am currently an advisor to seven Ph.D. students.

15. Before joining Pennsylvania State University as a professor, I was a software developer and project manager for companies in the networking industry including Applied Innovation, Inc., and Primary Access Corporation. I was also a senior researcher at AT&T Research-Labs. As part of my duties in these industrial positions, I informed, reviewed, and formed corporate policies and practices relating to the security of mobile devices, networks, and software systems (including filesystems).

16. I have published extensively in the field of mobile security, network and security management, computer systems, authentication, storage and filesystems, systems security, applied cryptography and network security. In addition to writing several articles for industry journals and conferences, I have authored portions of numerous books related to computer systems, applied cryptography and network security. I have served on the editorial boards of several peer-reviewed journals including ACM Transactions on Internet Technology, for which I was the Editor-in-Chief. I was also an Associate Editor for ACM Transactions on Information and System Security and IEEE Transactions of Software Engineering, two highly regarded journals in the field.

17. Additional information regarding my background, qualifications, publications, and presentations is provided in my CV, which I understand has been submitted as Exhibit 1004.

III. MATERIALS REVIEWED

18. In forming my opinions, I have reviewed the '338 Patent and considered each of the documents listed in the Exhibit List above and all other materials discussed herein. In reaching my opinions, I have relied upon my experience in the field and also considered the viewpoint of a person of ordinary skill in the art as of the time of the earliest claimed priority date of the '338 Patent, *i.e.*, May 5, 2009. As explained below, I am familiar with the level of a person of ordinary skill in the art regarding the technology at issue as of that time and all of the opinions I have formed have been assessed from the perspective of the person of ordinary skill in the art.

IV. DESCRIPTION OF THE RELEVANT TIMEFRAME, THE RELEVANT FIELD, AND A PERSON OF ORDINARY SKILL IN THE ART

19. I understand that the '338 Patent was filed on May 5, 2010, and claims priority to Provisional Application No. 61/175,489, filed on May 5, 2009. Thus, for purposes of my analysis, I have treated May 5, 2009 as the relevant priority date. I reserve the right to update my analysis should Patent Owner assert a different priority date.

20. I have received and understand the specification, claims, and file history of the '338 Patent.

21. In my opinion, a person having ordinary skill in the art ("PHOSITA") at the time of the effective filing date of the '338 Patent would have had at least a master's degree in computer science or electrical engineering or a related degree and with at least two years training or experience with networking and file systems. My opinions presented herein are as viewed through the eyes of a PHOSITA prior to May 5, 2009.

V. **TECHNOLOGY BACKGROUND**

A. **What are "File Systems"?**

22. A *file system* (commonly abbreviated as "fs") is the methods and data structures that an operating system uses to keep track of files on a storage device (e.g., hard disk, thumb drive). Ex-1031, p.1; *see also* Ex-1018. There are many kinds of different file systems, including minix, xia, ext3, ext2, jfs, xfz, zfs, msdos, vfat, among others. Ex-1031, pp.1-3. While the specific features provided by these varying kinds of file systems is broad (some of which I cover in subsequent sections of this report), they all perform the same basic function: to organize and keep track of files.

23. While some of the types of file systems above differ in some respects, there is a set of common structures and features that they all share (and broadly all

file systems available at the relevant time and today). These structures and features are core file system technologies; said differently, in the absence of these features, it could be argued that such technology could not be classified as a “file system.” These features include storage, filenames, directories (*i.e.*, folders), and metadata.

24. *Storage*, in the context of file systems, refers to components of a computer that are used to ensure that the data within a file system (and the computer broadly) is preserved when the system is shut down and restored upon start. Computer storage (*e.g.*, hard disk and solid-state drives) is typically characterized as *non-volatile*, in that the stored information is retained even after the system is shut down (and thus, power is removed). This is unlike computer memory (*e.g.*, RAM) which loses its information when the system shuts down. File systems are not limited to a single storage medium; they can, for example, be distributed across multiple storage mediums, wherein portions of the file system are “mounted” across multiple hard disk drives, for example.

25. The core feature of all file systems (available at the relevant time and today) is the notion of a *filename*, a collection of alphanumeric (and some special) symbols used in association with a collection of data. In most common file systems, filenames follow a name-dot-extension convention; filenames are preceded with a collection of symbols (often assigned by the user), a period, and an extension (typically assigned by an application). This convention allows files to be

recognizable by users (or other applications) and linked with a particular application so that when, for example, clicked on by a user, the appropriate application is opened and loads the contents of the file from storage into memory. For example, the filename “Document.docx” would be interpreted to be a Microsoft Word document.

26. Ostensibly all file systems (available at the relevant time and today) can be classified as, “hierarchical file systems”, wherein the file system organizes files into a tree-like structure. This organization is made possible by directories (sometimes also called “folders”), a structure which catalogues the contents inside of it. Directories can have directories inside of them (*i.e.*, subdirectories), sub-subdirectories within subdirectories, and so on. Often, the terms “parent” and “child” are used to describe the relationship between a directory (the “parent”) and a file or subdirectory inside of it (the “child”). In hierarchical file systems, files can be retrieved by navigating from the “root” parent directory (the directory in a file system that contains all files and subdirectories), through child subdirectories, and finally to the desired files. This process, navigating from parents to children to find files, is called a *file path* (or path). The path of a file describes both the location of a file within the file system and how the file should be organized when shown to the user (*e.g.*, as being “inside” of a folder). Technically speaking, file

paths are built from inode numbers, which inform the operating system where the data is physically located in computer storage.

27. Information such as file names and paths are stored alongside files as *metadata* (*i.e.*, data about data). This metadata is used by both the file system and operating system for a variety of functions, such as how files should appear to users (*i.e.*, their names), who owns the file (so that unauthorized users cannot access or modify file contents), their path, how much storage space the file consumes, *etc.* File system metadata can be read via the Unix system (a popular computer operating system whose origins date back to 1969) call “stat.” A system call (detailed in Section V.C) is a method by which applications can programmatically request a service from the operating system. Given a file name, stat returns information such as the current storage device the file is located on, its size, when it was created, among other pieces of information. Notably, this bookkeeping represents critical pieces of information for file systems; if metadata for a file were to be deleted, the file would be “lost” (in the sense that the operating system would not be aware of the existence of the file, which would require complex data recovery tools to rediscover the file on computer storage, if it were even possible).

B. The Evolution of File Systems

28. It is challenging to lay out the history of file systems since many of the earliest file systems were tightly coupled with the operating system (that is, it would be hard to treat the operating system and file systems as different entities, which is a paradigm that started in roughly the 1970s). Nonetheless, the evolution of files systems can be characterized through three epochs: migration from a flat to hierarchical file organization scheme, the ability to retrieve data remotely, and the seamless sharing of data across multiple devices.

1. From Files to Folders

29. Perhaps the earliest file system can be traced back to the DECtape (1964) and CP/M (1973) operating systems. The file systems inside of these operating systems organized files in a flat manner, i.e., there were no directories or folders to organize files within. This design was partially due to the storage capacities of computers at that time. While the systems did not exhibit any technical reason to be limited to a flat file organization, storage space was scarce, and thus, as a matter of practical limits, directories would not be particularly useful given the small number of files that could be saved to storage. However, as storage sizes increased, flat file systems rapidly became unwieldly from the sheer number of files applications, users, and operating systems would create.

30. To better managing an ever-growing number of files, the Multics operating system (Ex-1032), dating back to 1965, introduced hierarchical file systems. Multics influenced the design of many operating system available at the relevant time (*e.g.*, Unix), including its hierarchical file system design, support for long file names, storage quotas, *etc.* After Multics, essentially every popular file system implements a directory hierarchy, filenames, metadata, among other features. To support this organization schema, file metadata would be augmented to include where a file existed within the hierarchy, so that users and applications could focus on a desired set of files.

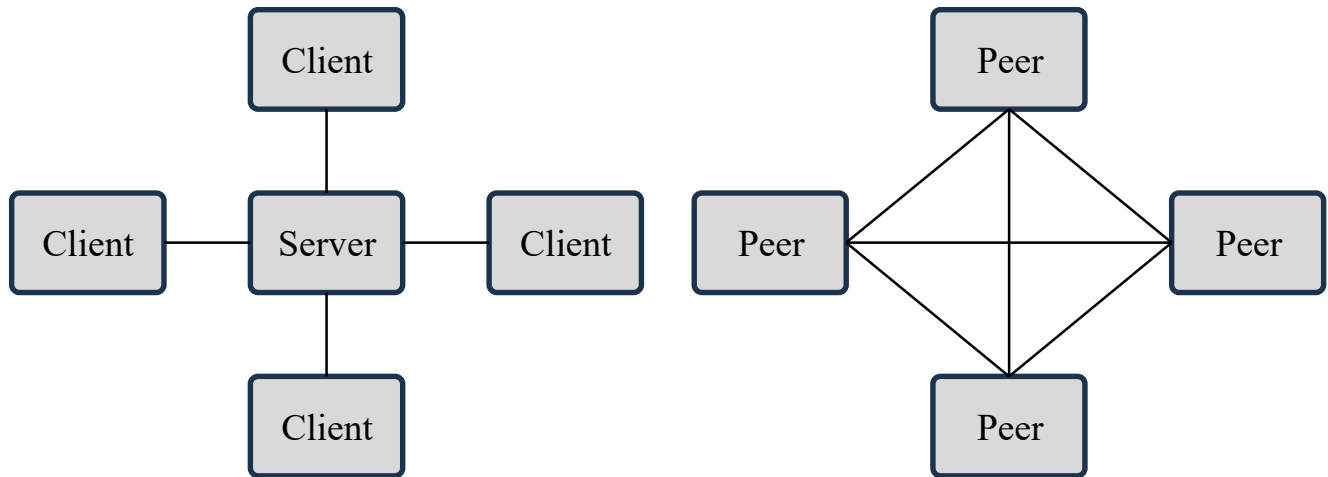
2. Remote Storage for File Systems

31. With the rise in popularity of computing systems and networks, file systems were further extended to allow access to storage over a network. In 1984, Sun Microsystems introduced the Network File System (NFS), a *distributed file system* that allowed users to access files stored remotely. Ex-1033. File systems using network protocols, such as NFS, are “distributed” in the sense that the methods to retrieve data are identical for both local and remote storage mediums. Thus, many distributed file systems are “transparent” in that users are not readily aware if files are retrieved via local or remote storage mediums. Specifically, such remote files are called *virtual files*, files whose content is stored remotely, with metadata such as file names, path, size, *etc.* stored locally. *See, e.g.*, Ex-1032,

12:32-13:11. These files support the transparency features provided by distributed file systems, in that virtual files appear structurally identical to all other local files in the file system.

32. While distributed file systems using simple network protocols offer many advantages, they are limited by: (1) files must be transferred over the network when a user or application uses them, which may be slow, and (2) the accessibility of files stored remotely is dependent on the connection to the remote computing device; a loss in connectivity implies users will no longer have access to the remote files (in other words, a single point of failure).

33. As networking technology improved, many computing technologies became increasingly reliant on accessing data remotely. The sheer amount of available data was problematic for computers available at the relevant time for at least two reasons: (1) storage mediums simply did not scale to the vast amount of desirable data (thus necessitating remote access via protocols such as NFS), and (2) the traditional client-server model, wherein multiple computers (clients) connected to a central computer (the server) for a particular service, became cost-prohibitive, as serving many clients mandated powerful (and expensive) computers to act as servers. One solution was to distribute work evenly (as much as possible) across clients, wherein clients would also serve the requests of others.



Client-Server vs. Peer-to-Peer Models

3. Sharing Files with Peers

34. This solution became known as *peer-to-peer* (P2P) computing. As the name implies, instead of “clients” and “servers”, all participating computers are considered “peers”, as shown in the Figure above. P2P protocols have been used in many applications, with the earliest popular application being Napster, a platform for sharing music among peers, in 1999. Ex-1035. Users would create accounts (which was most commonly done by visiting a website through a web browser, as exemplified by popular P2P technologies such as Napster) and advertise themselves to others to begin sharing data. Many file-system-like applications (such as PAST and IgorFs) rapidly emerged in the early 2000s with the popularity of P2P, offering the ability to store, retrieve, and use data much like classical distributed network protocols like NSF that relied on the client-server model.

35. While P2P applications offered many benefits over the traditional client-server model, they were not without limitation. Perhaps the largest assumption made in any P2P application was that all peers are accessible to all other peers (that is, even if two peers are across the planet, their traffic will eventually be routed to one another). As the Internet matured, this assumption was found to rarely hold true. New techniques for providing security to computers connected to the Internet, such as firewalls, network address translation (NAT), dynamic IP addresses, among others, implied an asymmetry broadly incompatible with P2P: a host that can connect to others does not necessarily mean that others can connect to the host. In this context, network address translation maps multiple private IP addresses into a single public IP address, allowing multiple devices in, *e.g.*, a home network to use a single IP address to access the Internet. Dynamic IP addresses change from time to time, which is problematic for P2P networks since IP addresses were used to identify peers (and were largely assumed to be static).

36. Clearly, this asymmetry assumption is detrimental to P2P protocols functioning properly. Thus, many P2P applications adapted to this maturation of the Internet to include networking techniques to allow computers to send data to one another even in circumstances where they cannot be directly connected to. Specifically, a *proxy server* is a computer that acts as an intermediary between other computers by relaying messages across computers. While proxies have many

uses (many of which are outside the scope of this report), in the context of P2P file system, a proxy server would allow, for example, two clients to send and receive files from one another even in circumstances where they may not be able to directly connect to one another (*e.g.*, firewalls, NAT, dynamic IPs, *etc.*). Proxies commonly used the *Hypertext Transfer Protocol (Secure)* (HTTP(S)), which is one of the most common protocols used across Internet. Ex-1016, Ex-1027. In the context of P2P networks, peers (who were unable to directly communicate with one another) would connect to the proxy server via HTTPS (just as they would when navigating the Internet) wherein communications between the peers would be relayed via the proxy server. This architecture is a hybrid between the traditional client-server model and peer-to-peer model in that the central server is leveraged only when a peer-to-peer connection is not possible. Leveraging proxies to address some of the inherent limitations of P2P applications was an obvious solution. Ex-1036.

C. The Fundamentals of File Systems

37. File names, directories, metadata, *etc.* are primitives that are found in nearly any file system (available at the relevant time and today). File systems implement a series of functions to support the use and organization of these data structures. That is, applications can call these methods to create, rename, move, open, close, *etc.* files within the computer as desired. The collection of these

methods is called the file system *application programming interface* (API). APIs are a core construct of computers in that they define the standards on how applications can communicate with one another.

38. Interfaces provided by the operating system (which includes file system interfaces) are called “system calls” (often abbreviated as a “syscall”). As the name suggests, a syscall is a particular service provided by the operating system for applications to use. These operating system interfaces are distinguished as: (1) applications can assume such interfaces will always exist for any collection of operating systems that implement the same standardized interfaces (*e.g.*, POSIX (Portable Operating System Interface), a set of standards which define the APIs for the operating system and filesystem in UNIX operating systems), and (2) commonly involve controlling hardware and thus, are thoroughly scrutinized for correctness (as an incorrectly implemented syscall could cause the operating system to stop functioning, thereby “crashing” the system). Consequently, software that modifies system calls fundamentally change the way the operating system/file system operates (discussed later in Section V.C.2).

1. The File System API

39. The purpose of a file system is to: (a) manage the locations of files and folders in non-volatile storage, (b) allow access to file data for applications to use. To support these basic features, all file systems (dating back to Multics)

provide at least the following interfaces: open, close, read, write, metadata management, and directory management. While there are other interfaces, these basic interfaces are commonly modified by software to provide new features. Notably, while references to documentation are for Linux (a popular, Unix-like operating system), all operating systems available at the relevant time (*e.g.*, Windows) provide file system interfaces that broadly provide identical functionality.

40. To “open” and “close” files are perhaps the most elementary concept in file systems. These are signals sent to the file system from applications that they wish to access data stored in storage. The file system then translates and sends these requests to the operating system to allocate the resources necessary (*e.g.*, reserving space in memory) to retrieve (or modify) the data at the locations recorded in the file metadata. The file is considered “open” when a data structure (commonly referred to as a “pointer”, “handle”, or “descriptor”) is returned from the file system to the application that contains the necessary information for the application to perform input or output operations on the desired file. Correspondingly, a file is considered “closed” when application notifies the file system that it wishes to terminate access to the file. On close, the operating system deallocates resources and the file system updates the metadata of the file.

41. When a file is open, applications can then either “write” or “read” data to or from the file, respectively. Reading data from a file involves, as the name describes, retrieving the data associated with the file, while writing involves modifying (or adding to) the file data. Both operations update the file metadata, in that writing data to a file would update “last file modification” metadata (and potentially the file size), while reading data from a file would update “last file access” metadata, among others.

42. As described in Section V.A, metadata records a variety of information about files, including their location on storage, their path within the file system hierarchy, event timestamps (such as opening, writing, *etc.*), among other pieces of information. In general, managing file metadata is the sole responsibility of the file system (in that file systems do not expose interfaces to modifying metadata, with exceptions for non-critical data, such as file timestamps). Instinctively, critical information such as the location of a file on storage should not be trivially modifiable by applications in that consequences of mistakes are severe (*e.g.*, the file could become “lost” to the operating system and thus unretrievable). If desirable, applications can also extend metadata structures to include arbitrary information.

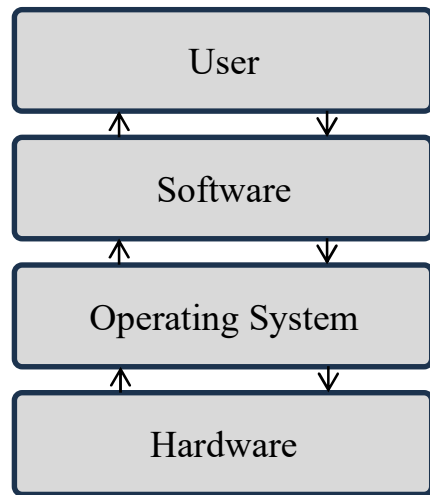
43. File systems also expose a series of interfaces for manipulating the path of a file in the filesystem hierarchy. Recall that the path of a file corresponds

to how the file is organized within directories (which is different from its physical location on storage). Applications can move files, change their names, or delete them.

2. Modifying File Systems

44. Since the 1960s, file systems have changed the way computers share and organize files. However, many of these changes do not require a developer to write their own file system from scratch when they wish to implement a new feature. Since the introduction of Unix and Windows 98, modifications done to file systems require simply writing and inserting the desired feature into the operating system. This capability of extending the features of file systems (and computers broadly) is one of the fundamental design philosophies of computers: the *principle of abstraction*. Exact definitions vary from application to application, but the principle of abstraction generally states that software should be designed so that its functions are independent of its implementation. In other words, developers that wish to extend or build “on top of” pieces of software should be able to focus on what the software does and not how it does it.

45. The principle of abstraction is embodied in the design of computers available at the relevant time and today; operating systems interface with the low-level functions of computer hardware to expose high-level features for the software above it to use, as shown in the Figure below.



The Abstraction Hierarchy

Here, computer hardware represents the least abstract layer of the computer hierarchy in that it (often quickly) performs a set of highly specific functions. The operating system sits above hardware, wherein it mediates access to hardware by providing a set of services for software to use (*e.g.*, a file system). Software sits above the operating system and interfaces with the computer to provide services for users to interact with. Notably, software need not be aware of precisely how the operating system manages the hardware—it only needs to be aware of the services that are provided by the operating system, and so on. Thus, it is common to describe software as sitting, for example, “below” the file system to mean that the software modifies the underlying services a file system interfaces with to perform its function.

46. Operating systems interface with hardware through *device drivers*, software that controls hardware and provides an interface for the operating system

to issue instructions to the underlying hardware. In Linux, device drivers are called loadable kernel modules (LKM) (Ex-1037), while in Windows they are called kernel-mode drivers. The kernel is the core component of the operating system that is responsible for managing system resources. To be precise, device drivers can run in user space (and thus, would be called “user-mode drivers” in Windows, for example). Without loss of generality, the file system device drivers in this declaration run in kernel-mode. In the context of file systems, device drivers could, for example, change the implementation of the write system call to exploit the design of the underlying storage medium (a notable example would be traditional hard disk drives, wherein data would be written in physically close clusters to optimize performance). Thus, modifying the file system involves rewriting the relevant system call implementations provided by operating systems.

47. Such file system modifications can offer a myriad of benefits (especially in the context of a distributed file system. Some examples include virus detection, automatic encryption, transparent replication, storage optimization, among other features.

D. Distributed File Systems in Peer-to-Peer Networks

48. A distributed file system in a peer-to-peer network relies on some form of a networking protocol (to retrieve data stored remotely), and retrieves data is from arbitrary computers in the network (that is, data is not exclusively provided

by a privileged set of computers, instead all computers in the network both provide and request data). These file systems all address the same fundamental challenges: (1) peer discovery (computers are aware of one another and the files that are part of the file system), and (2) data synchronization (file contents must be identical across peers). While proxies are not necessary in P2P networks, they are common in that they address many of the inherent limitations of P2P networks, as discussed in Section V.B.3.

49. The first step in any P2P network is joining the network. There are three approaches to joining a network. In “unstructured” networks, peers can periodically broadcast their presence, thus forming arbitrary connections to other peers who receive the broadcast. While easy to build, unstructured networks are highly inefficient in that queries for data must be flooded throughout the network, inducing substantial costs in traffic overhead. To address this inefficiency, “structured” networks organize peers such that file ownership is assigned to specific peers based on network quality criteria (*e.g.*, distributed hash tables). However, structured networks often struggle with peer discovery when peers are often joining and leaving the network. The final approach addresses both limitations via a hybrid approach; a centralized server maintains a list of peers while using approaches for efficient queries commonly found in structured networks.

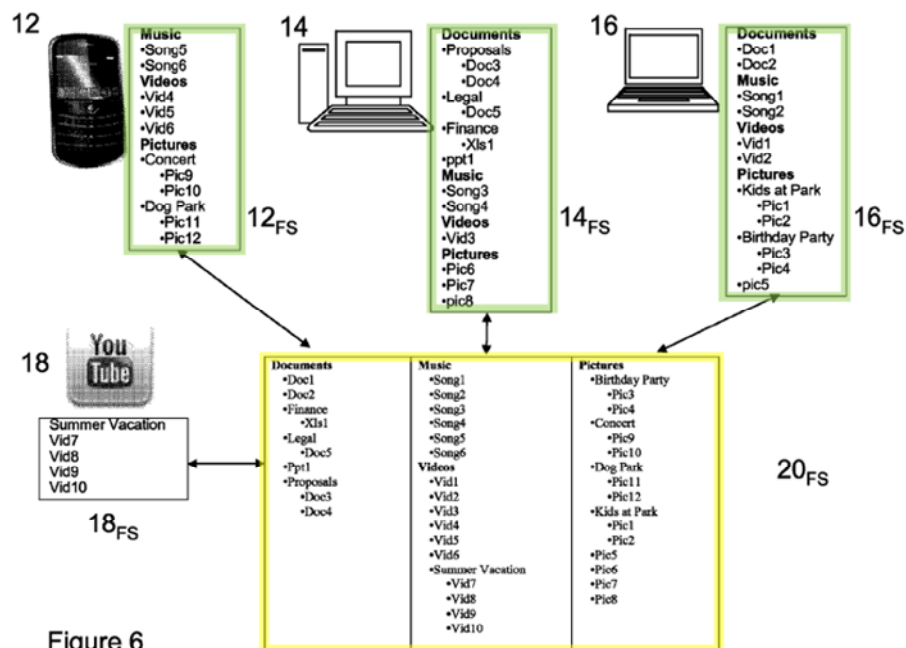
50. Once peers have joined the network, updates to files must be synchronized among peers. Data synchronization among peers is a well-known problem that contains multiple solutions, such as serialization (only one computer can update a file at a time), centralization (all updates go through one computer and relayed to others), and timestamps (only updates to the latest version of a file are valid). Each of these solutions have their own limitations in that they can require a central authority (thereby straining resources), disallow concurrent updates, or place limitations on the kinds of updates that are accepted (*e.g.*, only updates to the latest versions).

VI. OVERVIEW OF THE '338 PATENT

51. I have reviewed the '338 Patent and understand that Erik Caso and Michael Abraham are named as the inventors on this patent. The '338 patent relates to “[a] system and process that establish a singular shared file system among devices.” Ex-1001, Abstract. The system “create[s] a virtual representation of data on all of the devices via metadata indexing.” *Id.* For example, “[t]he system will differentiate data on a Local Device from data on a Remote Device” and “each client can create the virtual representation of the remote data within its local device file system.” *Id.*, Ex-1001, 6:14-18. “This allows for significant amounts of data to be virtually stored on each device, even if there is no physical storage room available for the data on each device.” *Id.*, Abstract.

52. The '338 patent discloses that the key to its purported invention is the creation and management of a “singular file system” with a “master index” for files on multiple user devices. *Id.*, Abstract; *see also id.*, 6:13-16 (“all of the individual meta-indices for all of the user’s registered devices [are integrated] into a database” and “[t]his **master index of metadata thus becomes the singular file system** that all devices use”).

53. Figure 6 of the '338 patent, to which I added coloring to facilitate understanding, illustrates “exemplary **unified file structure 20_{UFS}** [*i.e.*, the claimed master index, yellow] created from the individual device file structures 12_{UFS}, 14_{UFS} and 16_{UFS} [*i.e.*, the claimed individual indices, green] and on-line service file structure 18_{UFS}.” *Id.*, 9:13-18.



Ex-1001, Fig. 6. I note that although Figure 6 of the '338 patent illustrates 12_{FS}, 14_{FS} and 16_{FS}, the specification refers to them as 12_{UFS}, 14_{UFS} and 16_{UFS}, respectively. I also note that unless otherwise stated, I added the coloring to the figures in this declaration. Bolded text has also been added throughout my declaration for emphasis unless otherwise indicated.

54. I have also reviewed the prosecution history of the '338 patent. In the Applicant's response to the only Office Action, the Applicant amended claim 1, arguing that:

...the proposed combined teachings of Vesper, Wills, and Ben-Shaul fail to suggest the recited limitation of “providing by the at least one server the single master meta-index and meta-indices *for each of the other multiple devices to each of the multiple devices* via the individual software clients; at *each of the multiple devices, integrating metadata from the meta-indices of each of the other multiple devices into a local file system of each of the multiple devices to generate virtual files stored in the same locations as local files of the local file system, the virtual files indistinguishable from the local files by the local file system at each of the multiple devices.*” (emphasis added) as recited in claim 1, and similarly in claims 13 and 16.

Ex-1002, p.51.

55. To secure allowance of claims, the Applicant later agreed to an Examiner's amendment, which further narrowed claim 1 by requiring that "wherein the **individual software clients facilitate storage** of the data within each of the multiple devices **in accordance with the single meta-data index by modifying file systems** of each of the multiple devices **to include virtual files for data** from the single meta-data index that is **not local to a multiple device.**" Ex-1002, pp.16-17.

VII. OVERVIEW OF THE PRIOR ART

A. Aboulhosn

56. I understand that the Examiner was aware of Aboulhosn (Ex-1005) during prosecution. However, based on my review of the '338 patent prosecution history, I understand that the Examiner never applied Aboulhosn. I understand that Aboulhosn was cited by the Examiner during prosecution of the related '260 patent, but this occurred after the '338 patent claims were allowed. *See* Ex-1002, p.10 (Notice of Allowance dated September 12, 2012); Ex-1011, p. 177 (Non-Final Office Action dated October 25, 2012). Moreover, in response to the '260 patent's non-final Office Action dated October 25, 2012, the Applicant distinguished Aboulhosn from a limitation requiring that "virtual files appear indistinguishable to the singular file system." Ex-1010, claim 1. The Applicant distinguished Aboulhosn due to its disclosure of including ".vf" in the file name of virtual files.

Ex-1011, pp.54-55. The '338 patent claims include a similar “indistinguishability” limitation, and my opinions in this declaration do not rely on Aboulhosn for this limitation. Instead, my opinions rely on Phillips (which was not before the Office during prosecution of either the '260 or the '338 patents). The Aboulhosn-Phillips system does not include “.vf” or any other distinguishing characteristic in the file name.

57. In addition, none of the references relied upon in my declaration in combination with Aboulhosn (*i.e.*, Phillips, Shappell, and Tsang) were raised by the Examiner. Ex-1002, pp.60-69. As I mentioned above, the '338 patent was allowed after the Examiner added into the claims “wherein the individual software clients facilitate storage of the data within each of the multiple devices in accordance with the single meta-data index by **modifying file systems of each of the multiple devices to include virtual files for data from the single meta-data index** that is not local to a multiple device.” *Id.*, pp.15-16. This, in my opinion, was well known in the art, as evidenced by Aboulhosn, Phillips, and Shappell (which I will discuss below), none of which were raised by the Examiner in an office action during prosecution. Therefore, in my opinion, the grounds presented here are material and not cumulative of previous art considered by the Examiner. Therefore, in my opinion, Aboulhosn is material to the patentability of at least some of the challenged claims of the '338 patent.

B. Phillips

58. I understand that the Examiner was not aware of Phillips (Ex-1006) during prosecution.

59. Like the '338 patent, Phillips discloses “a multi-user file storage service and system” where “[e]ach user of [a] user group is enabled to access the files of the file group via the respective client node in communication with the remote file server node via the wide area network.” Ex-1006, Abstract; 6:4-7. Moreover, “the virtual storage device, and its contents (i.e., all of the files, directories/folders, etc. stored in the virtual storage device), appears, to the client node user and the application executing at the client node, to be locally present.” *Id.*, 14:62-66. When a user needs to access a remote file on the virtual storage device, “the client node software intervenes and assists the operating system in identifying the appropriate file and in providing the data of the file to the operating system to complete the access to the file (i.e., read, write, modify, delete, etc.).” *Id.*, 14:54-58.

C. Shappell

60. I understand that the Examiner was not aware of Shappell (Ex-1007) during prosecution.

61. Shappell “provides a method for creating a shared space, and enables sharing of files in the shared space.” Ex-1007, ¶ 9. Like the '338 patent, the

“shared space is presented through a graphical user interface with the look and feel of existing file system features.” *Id.*, ¶ 43. According to Shappell, “the invention may be implemented as a file system driver or filter driver that resides below the file system and accesses file system services provided by an operating system.” *Id.*, ¶ 44.

D. Tsang

62. I understand that the Examiner was not aware of Tsang (Ex-1008) during prosecution.

63. Like the '338 patent, Tsang also “provides a combined file space for files from multiple partner computing devices.” Ex-1008, Abstract. For example, as shown in Tsang’s Figure 1 below, “[e]ach partner computing device 102-110 includes a respective combined file space.” *Id.*, 3:28-29.

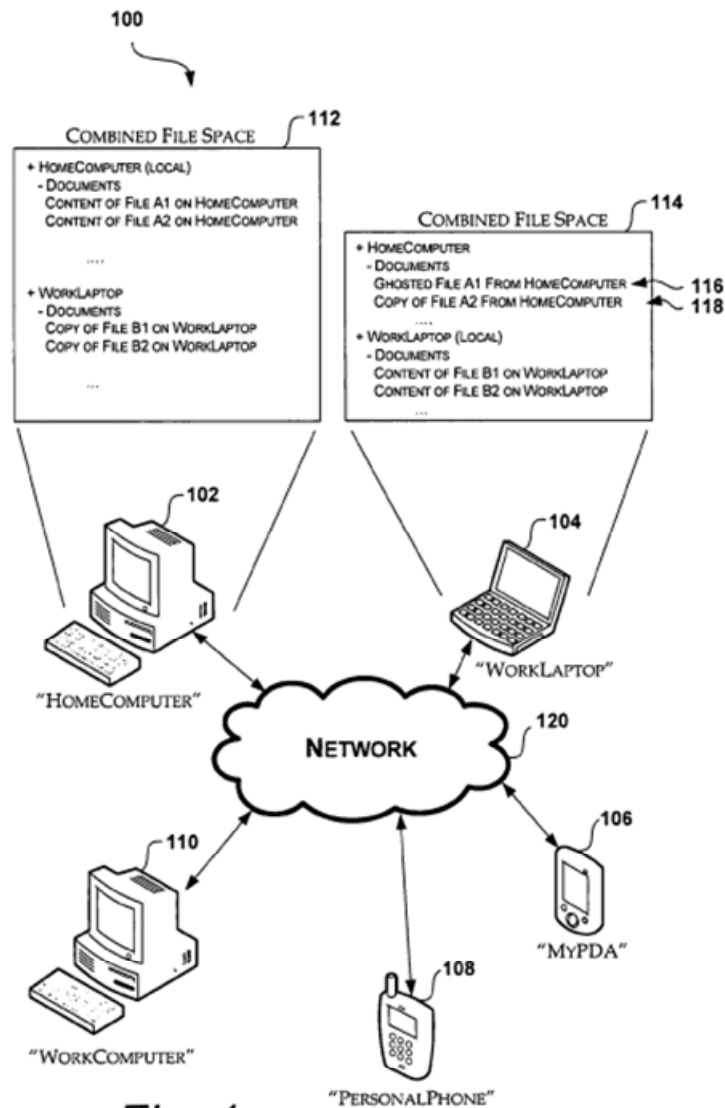


Fig. 1

Ex-1008, Fig. 1. Moreover, also like the '338 patent, “[i]n order to provide the same view of the shared files in a combined file space with limited space, the combined file space may include a “ghosted file” (e.g., ghosted file 116) instead of actual content (e.g., content 118) for the associated file.” *Id.*, 3:43-47.

VIII. CLAIM CONSTRUCTION

64. I understand that claim construction is the process of determining the meaning of a term or phrase in a patent claim. I understand that the proper construction of a term is how a PHOSITA would have understood the term based on its use in the claims, specification, and file history of the patent. I further understand that the claims are construed before the Board according to the same claim construction standard that applies in district courts. I have followed these principles in my analysis throughout this declaration.

A. Terms to Be Construed

1. “web-based system”

65. Independent claim 1 requires a “web-based system.” The term “web-based” is not explicitly defined in the specification, but it is my opinion that a PHOSITA would have understood that the “web” refers to World Wide Web, which is “a system of interlinked hypertext documents accessed via the **Internet**.” Ex-1026, p.1. In other words, a PHOSITA would have recognized that a “web-based system” is a system that operates on the **Internet** to provide a service. It uses Internet protocols such as FTP or HTTP and is commonly accessed by a user via a web-browser. This understanding is consistent with the ’338 patent specification: “Web Service—is a part of the system that operates on the **Internet**...” Ex-1001, 4:43-44.

66. To the extent PO argues that “web-based” requires using a web protocol, such as http, and/or using a web browser, I disagree with such a narrow interpretation is appropriate. *See* Ex-1001, 7:6-7 (“the web service will itself manage the connection (via HTTP Relay) between devices”); 7:10-11 (“a browser window is opened within the on-line service to facilitate operation”). Nevertheless, in my opinion, this too is taught by the prior art.

67. As I will explain in more detail below, Aboulhosn discloses a system that communicates via the Internet, *i.e.*, a web-based system. To the extent a narrower construction of the term “web-based system” is adopted, Phillips discloses it. *See* Section XII.A.1.b.

68. For purposes of this declaration, I do not believe that any other express constructions are needed and have applied the plain and ordinary meaning of all terms.

IX. ANTICIPATION

69. I have been instructed as to the definition of “anticipation” in the context of the patent laws.

70. I understand that anticipation of a claim requires that every element of a claim be disclosed expressly or inherently in a single prior art reference, in combination, as claimed. I understand that a single prior art reference may anticipate claims without expressly disclosing a feature of the claimed invention if

that feature is necessarily present, or inherent, in that reference. I understand that a reference is read from the perspective of one of ordinary skill in the art at the time of the invention.

X. OBVIOUSNESS

71. I have been instructed as to the definition of “obviousness” in the context of the patent laws.

72. It is my understanding that obviousness is a question of law based on underlying factual issues including (1) the scope and content of the prior art, (2) the differences between the prior art and the challenged claims, (3) the level of ordinary skill in the pertinent art, and (4) the existence of secondary considerations such as commercial success, long-felt but unresolved needs, failure of others, copying, industry praise, etc.

73. It is also my understanding that I should first determine the scope and content of the prior art, then consider the differences, if any, between the prior art and each challenged claim; and that in doing so, although it is proper to consider differences between the claimed invention and the prior art, I should not focus on only the differences, because the test is whether the claimed invention as a whole would be obvious over all of the prior art. It is also my understanding that, in evaluating validity, each claim of the patent must be considered separately and in its entirety.

74. It is my understanding that the next step is to determine the level of ordinary skill in the art to which the claimed invention pertains at the time the invention was made.

75. It is my understanding that a PHOSITA is presumed to have knowledge of the relevant prior art at the time of the claimed invention; and that if I find that the available prior art shows, discloses, teaches or suggests each of the elements of the challenged claims, I should determine whether a PHOSITA would have had a reason to combine or coordinate these elements in the same manner as the challenged claim.

76. I understand that an obviousness evaluation can be based on a combination of multiple prior art references.

77. I understand that for a single reference or a combination of references to render obvious the claimed invention, a PHOSITA must have been able to arrive at the claimed invention by altering or combining the applied reference or references. I also understand that obviousness arises where it is determined that the claimed invention implemented a predictable variation of prior art elements or amounts to nothing more than the predictable use of prior art elements according to their established functions.

78. I understand that the prior art references themselves may provide an explicit suggestion, motivation, or reason to combine, but that the suggestion to

combine may also arise from the state of the prior art generally. I also understand that when considering the obviousness of a patent claim, one should consider whether a teaching, suggestion, or motivation to combine the references exists to avoid impermissibly applying hindsight when considering the prior art. I understand this test should not be applied rigidly, but that the test can be important to avoid such hindsight.

XI. OPINIONS REGARDING PATENTABILITY

79. In my opinion, claims 1-2, 5-6, 7-9, 10-14 are rendered obvious by Aboulhosn in view of Phillips. Additionally, claims 3 and 4 are rendered obvious by the Aboulhosn-Phillips combination further in view of Tsang.

80. Moreover, in my opinion, claims 1-2, 5-6, 7-9, 10-14 are also rendered obvious by Aboulhosn in view of Phillips and Shappell. Additionally, claims 3 and 4 are rendered obvious by the Aboulhosn-Phillips-Shappell combination further in view of Tsang.

81. I have based my opinion on the following references:

- **Aboulhosn:** U.S. Patent No. 6,938,042 titled “Peer-to-peer file sharing” to Amir L. Aboulhosn *et al.* (“Aboulhosn,” Ex. 1005), was filed October 8, 2002, published April 8, 2004, **issued August 30, 2005**. I understand that Aboulhosn is prior art to the ’338 patent.

- **Phillips:** U.S. Patent No. 7,058,696 titled “Internet-based shared file service with native PC client access and semantics” to Robert S. Phillips *et al.* (“Phillips,” Ex. 1006), was filed November 1, 2000 and **issued June 6, 2006**. I understand that Phillips is prior art to the ’338 patent.

- **Shappell:** U.S. Patent Publication No. 2005/0091595 titled “Group shared spaces” to Michael Shappell *et al.* (“Shappell,” Ex. 1007), was filed October 24, 2003 and **published April 28, 2005**. I understand that Shappell is prior art to the ’338 patent.

- **Tsang:** U.S. Patent No. 7,930,270 titled “Managing files on multiple computing devices” to Michael Tsang (“Tsang,” Ex. 1008), was **filed February 26, 2007**, published on August 28, 2008, and issued April 19, 2011. I understand that Tsang is prior art to the ’338 patent.

82. In my opinion, the challenged claims of the ’338 patent are unpatentable based on the following grounds:

Ground	Statute	Reference(s)	Claims
1	§103	Aboulhosn, Phillips	1-2, 5-6, 7-9, 10-14
2	§103	Aboulhosn, Phillips, Tsang	3-4
3	§103	Aboulhosn, Phillips, Shappell	1-2, 5-6, 7-9, 10-14

Ground	Statute	Reference(s)	Claims
4	§103	Aboulhosn, Phillips, Tsang, Shappell	3-4

XII. MY OPINIONS AS TO THE CHALLENGED CLAIMS OF THE '338 PATENT

A. My Opinions as to Claims 1-2 and 5-14 in view of the Aboulhosn-Phillips combination

1. Claim 1

83. In my opinion, claim 1 is unpatentable because it is rendered obvious by Aboulhosn in view of Phillips. Claim 1 of the '338 Patent recites the following.

I address each of these limitations in my analysis below.

1[P]: A process for establishing a singular file system across multiple devices comprising:

1[A]: receiving user information to open an account for establishing a singular file system across multiple devices via a web-based system that includes at least one server;

1[B]: installing an individual software client on each of the multiple devices via the web-based system;

1[C]: accepting registration of multiple devices via the web-based system;

1[D]: scanning each of the multiple devices by each of the individual software clients to inventory data on each of the multiple devices and create a meta-index of the files for the inventoried data;

1[E]: providing by the individual software clients via the multiple devices individual meta-indices of

the inventoried data for each of the multiple devices to the at least one server;

1[F]: integrating by the at least one server the individual meta-indices to create a single master meta-index;

1[G]: providing by the at least one server the single master meta-index and meta-indices for each of the other multiple devices to each of the multiple devices via the individual software clients;

1[H]: integrating metadata from the meta-indices of each of the other multiple devices into a local file system of each of the multiple devices to generate virtual files stored in the same locations as local files of the local file system, the virtual files indistinguishable from the local files by the local file system at each of the multiple devices; and

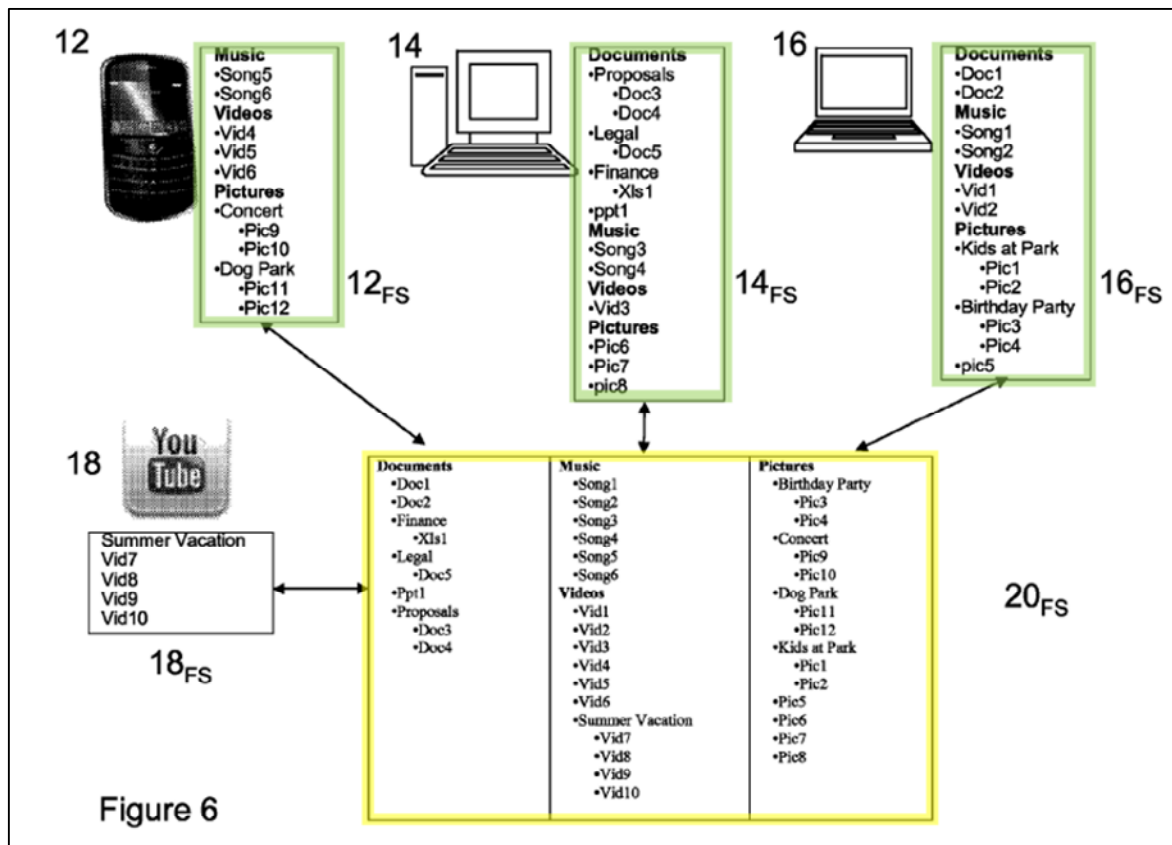
1[I]: continually updating the single master meta-index on the at least one server and each of the multiple devices in response to changes to the data indexed thereon,

1[J]: wherein the individual software clients facilitate storage of the data within each of the multiple devices in accordance with the single meta-data index by modifying file systems of each of the multiple devices to include virtual files for data from the single meta-data index that is not local to a multiple device.

a. 1[P]

84. If the preamble is considered a limitation of the claim, it is my opinion that Aboulhosn discloses it.

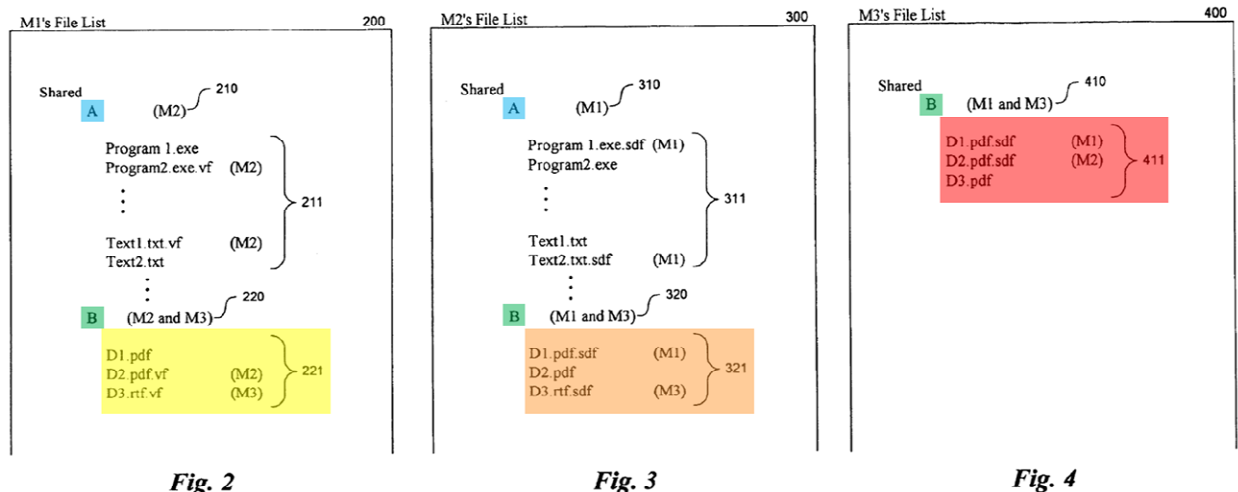
85. The '338 patent describes “a singular file system” as a system used to manage and display files across multiple devices. *See, e.g.*, Ex-1001, 3:17-19, 5:63-64, 6:13-16. As shown in Figure 6 below (to which I have added colored annotations), the '338 patent illustrates its singular file system using an “exemplary unified file structure 20_{UFS} [yellow] created from the individual device file structures 12_{UFS}, 14_{UFS} and 16_{UFS} [green] and on-line service file structure 18_{UFS}” and “[t]his unified file structure 20_{UFS} [yellow] will replace file structures 12_{UFS}, 14_{UFS} and 16_{UFS} in devices 12, 14 and 16.” *Id.*, 9:13-18.



Ex-1001, Fig. 6.

86. In my opinion, Aboulhosn discloses the same. For example, Aboulhosn discloses “[a] method and system for **sharing files between a group of computer systems**” where “[t]he files shared by a group are associated with a group folder.” Ex-1005, 2:1-2. Aboulhosn’s “group folder is represented by a folder created by the file system of each member of the group” and “[t]he folder at each member contains a file for each file that is shared by the group.” *Id.*, 2:3-5. In other words, a PHOSITA would have recognized that Aboulhosn discloses establishing a file system (including the shared/group folder) common to multiple computer systems.

87. Also, Aboulhosn’s Figures 2-4 (as I have annotated below) illustrate a unified file structure shared between three computer systems M1, M2, and M3, each with the same shared folder. Ex-1005, 4:15-16.



As shown, “M1 and M2 are members of a group ‘A’ [blue] and M1, M2, and M3 are members of group ‘B’ [green].” *Id.*, 4:16-17. Each of M1, M2, and M3 share a unified file structure—*i.e.*, 221 for M1 (yellow), 321 for M2 (orange), and 411 for M3 (red). Like the claimed “singular file system,” Aboulhosn’s unified file structure also includes shared files from individual devices—*i.e.*, “D1.pdf” from M1, “D2.pdf” from M2, and “D3.pdf” from M3.

88. Therefore, in my opinion, Aboulhosn discloses “[a] process [described in subsequent steps] **for establishing a singular file system** [e.g., Aboulhosn’s file system including shared/group folder(s)] **across multiple devices** [e.g., group members, such as M1, M2, and M2].”

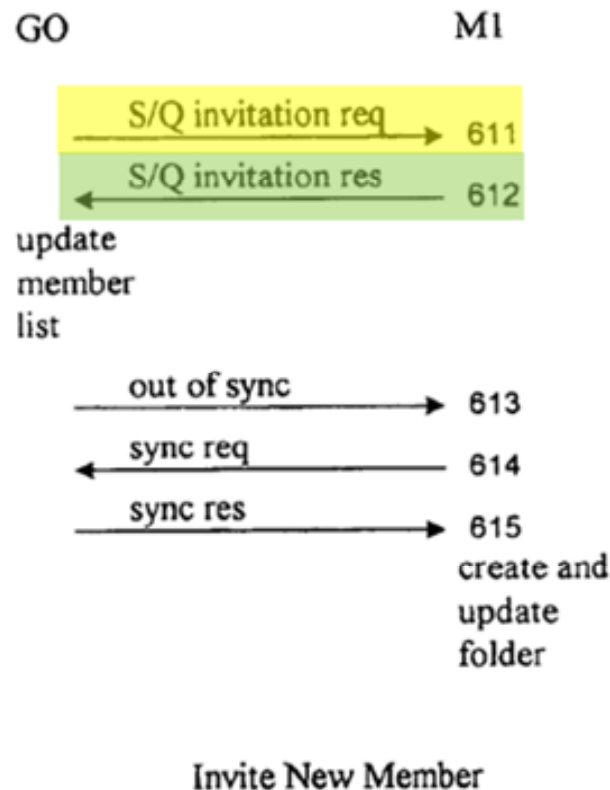
b. 1[A]

89. Limitation 1[A] requires: “receiving user information to open an account for establishing a singular file system across multiple devices via a web-based system that includes at least one server.” I will address different aspects of this limitation in turn below.

90. **Receiving user information.** Aboulhosn discloses “a group owner [“GO”] component that **creates a group**...[and] invites computer systems to join the group as members....” Ex-1005, claim 39. In my opinion, Aboulhosn discloses that in creating the group of authorized members to establish

Aboulhosn's singular file system, user information is received by Aboulhosn's system.

91. For example, Aboulhosn's "group owner [*i.e.*, GO computer] initially receives an indication from its user [*i.e.*, GO person] that **a registered computer system is to be invited** to join the group." *Id.*, 6:8-10. "The group owner [*i.e.*, GO computer] then sends an invitation request message 611 [yellow] to the invited computer system." *Id.*, 6:13-15. "When the invited computer system receives the invitation request message, it notifies its user [e.g., invited member device's user] of the invitation request." *Id.*, 6:37-39.

**Fig. 6A**

Ex-1005, Fig. 6A. According to Aboulhosn, the invited computer system accepts the invitation by “send[ing] **an invitation response message 612** [green] to the group owner [*i.e.*, GO computer]... [who] updates its member list for that group accordingly.” *Id.*, 6:41-44. In other words, a PHOSITA would have recognized that to create an account (*e.g.*, an established group of authorized users), Aboulhosn’s GO receives user information, such as an identification of the to-be-invited device and a response message accepting an invitation.

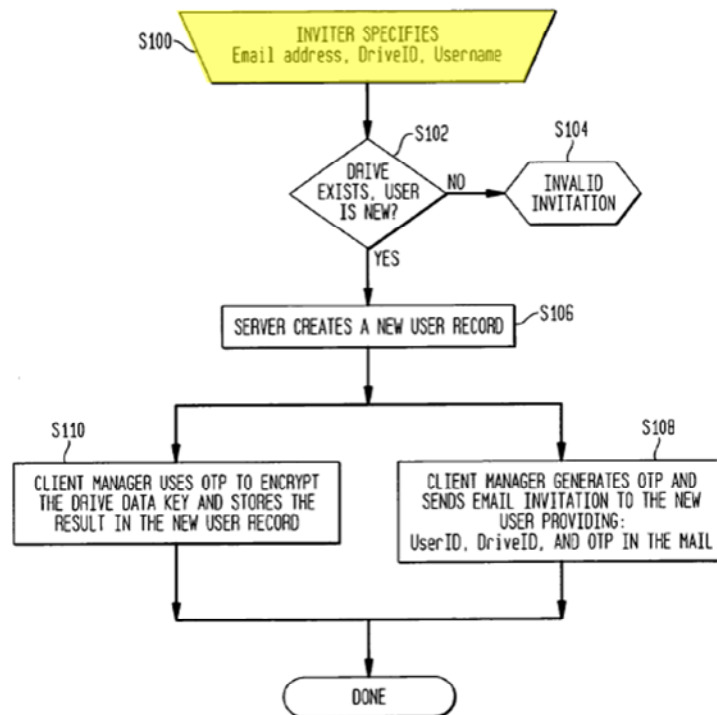
92. Aboulhosn also discloses that “[a] member comes online by providing **authentication information** to the authentication server.” Ex-1005, 7:33-35. The ’448 Application—from which Aboulhosn is a continuation-in-part, claims priority, and “incorporate[s] by reference in its entirety” (Ex-1005, 1:5-8)—explains that authentication information includes “**a user name and password.**” Ex-1009, p.6, 4:6; p.9, 7:12-16; p.12, 10:4-6. Thus, in my opinion, a PHOSITA would have understood that when creating an account (*i.e.*, an established group of authorized users), Aboulhosn’s authentication server receives **authentication information, such as usernames and passwords** (just like in the ’338 patent), to store in its “registered table” for later authentication. Ex-1005, 5:33-38; Ex-1001, 5:39-42 (“Registration information includes **user name**...as well as creation of **user name and password.**”).

93. In fact, in my opinion, it was well known to a PHOSITA that when creating an account, corresponding user information, such as user name and password, is received by the relevant server from, for example, a user. *See, e.g.*, Ex-1019, 1:58-21 (“consumers redundantly register numerous pieces of vital **account information** with various companies each day,... includ[ing] actual user’s name, an account username, address, social security number, telephone numbers, email addresses, and even personal profile information such as gender, birth date, brand preferences, vocation, hobbies, food preferences, etc. Many sites

also... allow[s] the user to select a password within some constraints, such as minimum character length.”).

94. To the extent PO argues that Aboulhosn does not disclose “receiving user information,” it is my opinion that Phillips discloses it. In particular, Phillips discloses “[a]ccepting requests from client manager nodes to **create client user accounts.**” Ex-1006, 12:49-50. Phillips’s “client manager node is able to designate new user accounts” by “issu[ing] a message containing a command to invite a new user, **the email address of the new user, a user name for the new user and an identifier of the virtual storage device (‘drive id’) of the virtual storage device** on which the new user is to be invited.” Ex-1006, 12:36-37, 16:24-28; *see id.*, Fig. 5 (step S100, **yellow**).

FIG. 5



Ex-1006, Fig.5. In my opinion, a PHOSITA would have understood that in issuing invitations, Phillips’s client manager node necessarily receives (*e.g.*, from the user) user information, such as email addresses, usernames, and an identification of user devices. *See* Ex-1001, 5:40-41 (“Registration information includes **user name**, **e-mail address**....”).

95. **Web-based system.** As I discussed in Section VIII.A.1, a “web-based system” is a system that operates on the **Internet** to provide a service using Internet protocols, such as FTP or HTTP, and is commonly accessed by a user via a web-browser. In my opinion, Aboulhosn discloses this limitation because

Aboulhosn's "computer systems 510 communicate with each other and with authentication server 520 via **communications link 530**," including "the **Internet**, a local area network, a **wide area network**." Ex-1005, 4:67-5:2, 5:58-60.

Therefore, Aboulhosn's system is a system that operates on the Internet to provide services, such as file synchronization and account registration.

96. Aboulhosn also discloses use of an Internet protocol, such as FTP. *Id.*, 1:24-25. Since FTP was one of the most common protocols at the time used in web-based systems, it would have been obvious to a PHOSITA to use a web-based Internet communications link in Aboulhosn. *See, e.g.*, Ex-1044 and Ex-1045. Moreover, at the time of the '338 patent, it was common to register an account by visiting a website through a web browser. *See, e.g.*, Ex-1041, Ex-1046. Therefore, a PHOSITA would have found it obvious to implement Aboulhosn's account registration via a web-based system to address Aboulhosn's "unfamiliar user interface problem." Ex-1005, 1:25-30.

97. To the extent that Patent Owner argues that Aboulhosn does not disclose a "web-based system" requiring using a web protocol (such as http) or a web browser, it is my opinion that Phillips discloses it. *See* Ex-1006, 8:9-14 ("Communication is achieved in the **Internet** using a hierarchy of protocols, including...the **hypertext transfer protocol (http)** and the **hypertext transfer protocol over secured sockets layer (https)**.'").

98. Phillips also discloses that it was well known that a “user can access [a remote storage device] while executing a **web browser program** on the user’s computer terminal, to store data for later retrieval.” Ex-1006, 2:41-44.

99. As I will explain in more detail in Section XII.A.2, a PHOSITA would have been motivated to incorporate Phillips’s http/https protocol and web browser into Aboulhosn’s system. Therefore, in my opinion, the Aboulhosn-Phillips system is the claimed web-based system.

100. **At least one server.** In my opinion, it is well known that the plain and ordinary meaning of a server is “a computer in a network that is used to provide services (as access to files or shared peripherals or the routing of e-mail) to other computers in the network,” *e.g.*, clients. Ex-1020, pp.3-4. This is what Aboulhosn’s GO computer does—GO “ha[s] the sole authority to invite other computer systems to join the group,” “maintains a list of the group members,” and routes “messages related to metadata updates.” Ex-1005, 1:63-65, 2:42-43, 2:43-44. For example, “[w]hen a member goes online, it contacts the group owner (assuming that the group owner is currently online) to synchronize its copy of the group folder with the group owner.” *Id.*, 2:64-67. In other words, a PHOSITA would have recognized that Aboulhosn’s GO provides device registration, group member maintenance, metadata routing, and file synchronization services to member devices, *e.g.*, clients, and is therefore a “server.”

101. I note that this limitation recites “receiving user information...via a web-based system that includes at least one server,” which does not require specifically that the “at least one server” (*i.e.*, Aboulhosn’s GO computer) receive user information. Aboulhosn teaches that “[t]he functions of the authentication server may be divided among multiple servers.” Ex-1005, 5:46-49. Thus, in my opinion, a PHOSITA would have understood that Aboulhosn’s GO computer can be a separate and distinct server or can be combined with at least some functions of the authentication server, including receiving user information.

102. Therefore, in my opinion, Aboulhosn in view of Phillips discloses **“receiving user information to open an account for establishing a singular file system** [*e.g.*, Aboulhosn’s GO creating the group from information received from the GO user and also receiving information from individual member users, including usernames and passwords to establish a group of authorized users, or Phillips’s creating accounts using received username and email address] **across multiple devices** [*e.g.*, Aboulhosn’s member devices] **via a web-based system** [*e.g.*, Aboulhosn’s system, or Aboulhosn’s system using Phillips’s http/https for network communications and/or web browser] **that includes at least one server** [*e.g.*, Aboulhosn’s GO computer acting as a server].”

c. 1[B]

103. Limitation 1[B] requires: “installing an individual software client on each of the multiple devices via the web-based system.”

104. The '338 patent states that a “Software Client (or the Client)—is a piece of software that is installed and runs continuously on the device.” Ex-1001, 4:41-42; *see id.*, 5:43-46.

105. In my opinion, Aboulhosn discloses the same. Aboulhosn discloses that each “computer system[] 510 include[s] **a peer component 511**...which may be **downloaded** from the authentication server when the computer system is registered, [that] controls the sharing of files with other computer systems.” Ex-1005, 4:66-5:9. Peer component 511 (blue) is depicted in Figure 5, below.

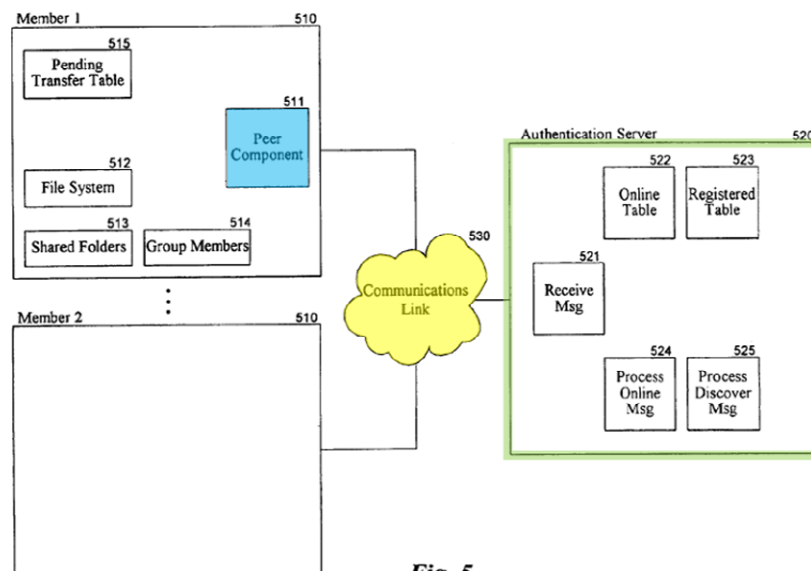


Fig. 5

Ex-1005, Fig. 5. Aboulhosn explains that “the functions of **the peer component**...include registering a computer system, placing a computer system online, creating groups, inviting computer systems to become members of a group, sending updates to metadata, and so on.” *Id.*, 5:11-14; *see also id.*, 7:65-8:67, Figs. 7-10. Thus, it is my opinion that Aboulhosn’s peer component 511 is a software client downloaded from the authentication server 520 and installed on each of Aboulhosn’s member devices.

106. In my opinion, Aboulhosn also discloses that its software client (*i.e.*, peer component) is installed “via the web-based system.” Aboulhosn’s peer component is “**downloaded** from the authentication server [**green**]” via communications link 530 (**yellow**). Ex-1005, 5:6-8. Aboulhosn’s Figure 5 below clearly shows that the peer component 511 (**blue**) is installed on member devices.

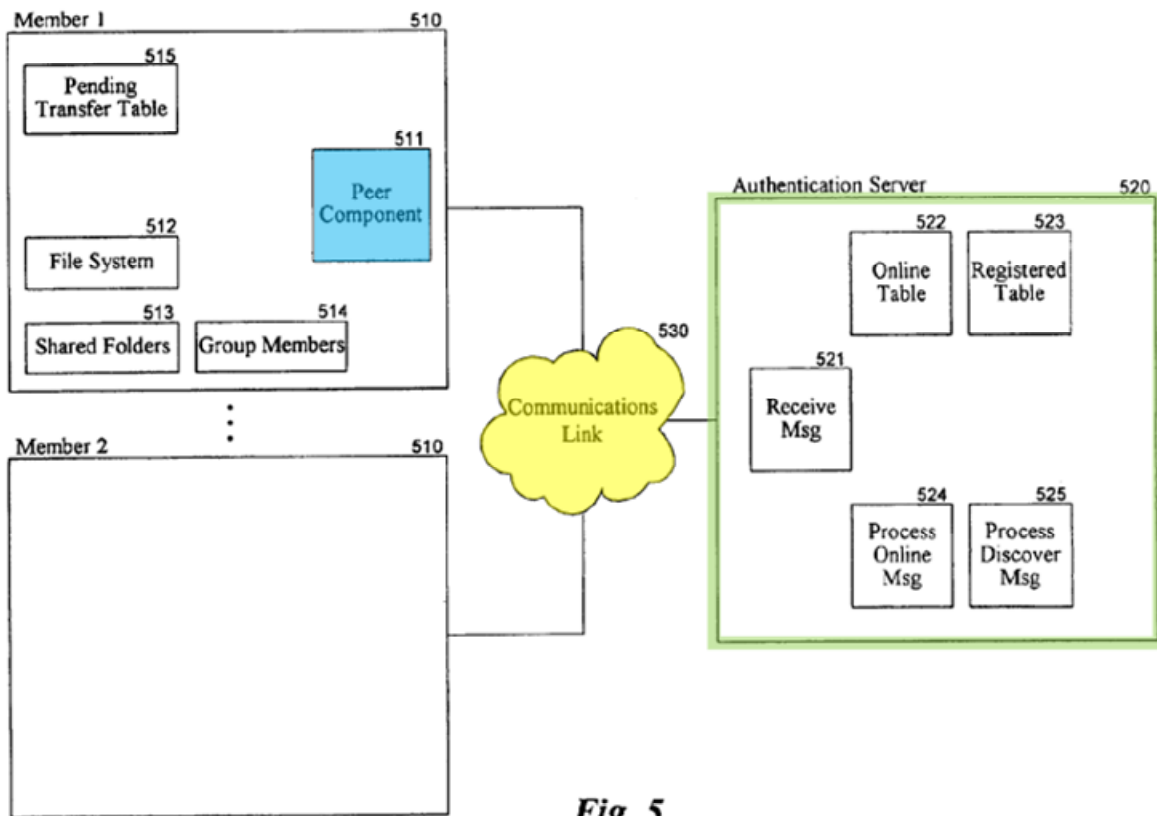
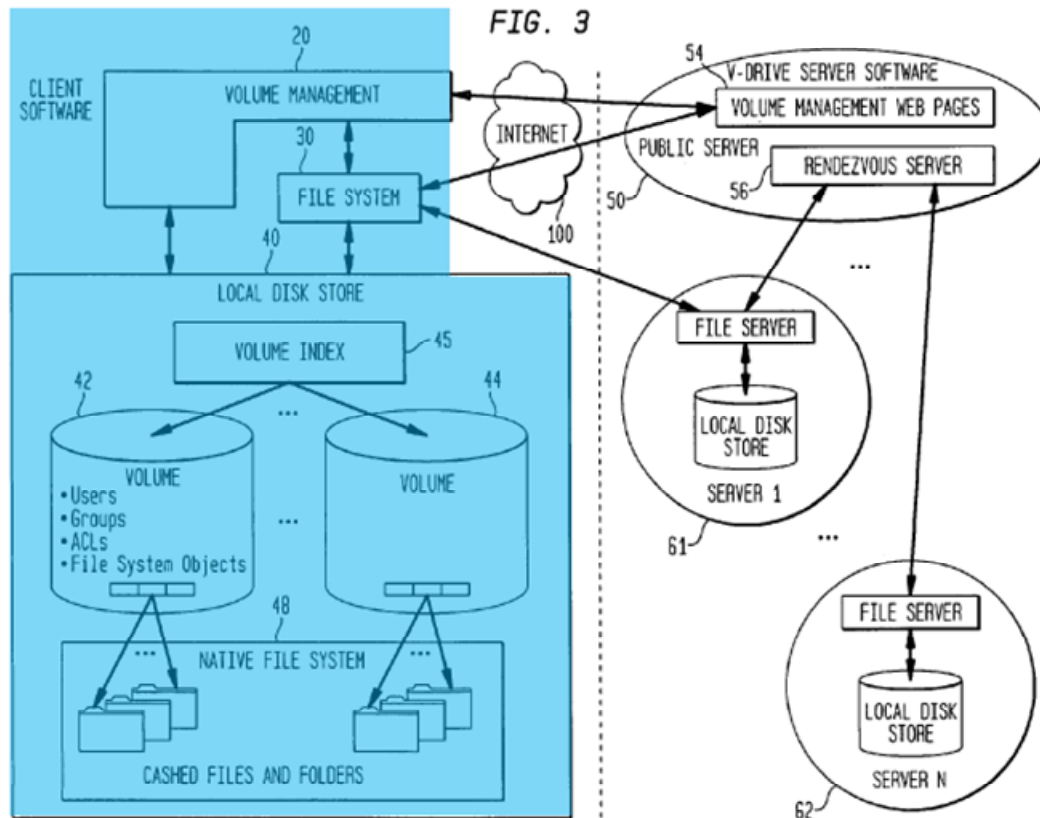


Fig. 5

Ex-1005, Fig. 5. In my opinion, a PHOSITA would also have understood that for Aboulhosn’s peer component to operate and perform various functions as described, it is necessarily installed on member devices. As I have explained above, Aboulhosn’s communications link is an Internet connection (*i.e.*, a web-based system), and, therefore, its peer component is installed via the web-based system. *See* Section XII.A.1.b.

107. To the extent PO argues that Aboulhosn does not disclose installing an individual software client via a web-based system, Phillips discloses it. Phillips discloses a client node software architecture (blue) in Figure 3 below, which “can

be **deployed** on moveable computer terminals (such as laptops) and or computer terminals.” Ex-1006, 11:19-24.



Ex-1006, Fig. 3.

108. A PHOSITA would have understood that “deploying” here means “installing” a software client. *See, e.g.*, Ex-1038, p.1 (“**System deployment**—ordering, receiving, and **installing the new system hardware and software**—is one of the most exciting and stressful parts of your job.”); Ex-1039, p.4 (“**Software deployment involves** assembling and **installing** software distributions or packages: a collection of files and directories containing programs, libraries,

documentation, etc.”). In fact, it is my opinion that installation of client software for file synchronization was well known before the ’338 patent. *See, e.g.*, Ex-1022, 3:52-56 (“a stand-alone application may be **installed** and stored upon the user’s computer. This stand-alone application serves to intermediate the user commands with the web server and ultimately the metadatabase in the Internet Storage device.”). As I explained above, Phillips also describes its system is web-based.

109. Therefore, in my opinion, Aboulhosn in view of Phillips discloses “**installing an individual software client** [*e.g.*, installing Aboulhosn’s downloaded peer component or Phillips’s client software architecture] **on each of the multiple devices** [*e.g.*, Aboulhosn’s member devices M1, M2, and M3] **via the web-based system** [*e.g.*, Aboulhosn’s system, or Aboulhosn’s system using Phillips’s http/https for network communications and/or web browser].”

d. 1[C]

110. Limitation 1[C] requires “accepting registration of multiple devices via the web-based system.” In my opinion, Aboulhosn discloses this limitation.

111. Aboulhosn discloses that “a computer system is eligible to be a member of a group after **the computer system has registered with the authentication server.**” Ex-1005, 3:15-18. As I annotate below, Aboulhosn’s “authentication server [**green**] includes...a registered table 523 [**purple**]...[that]

contains an entry for each computer system that is currently **registered with the file sharing system.**” Ex-1005, 5:27-35; 5:46-49 (“[T]he authentication server...**handl[es] registration requests....**”).

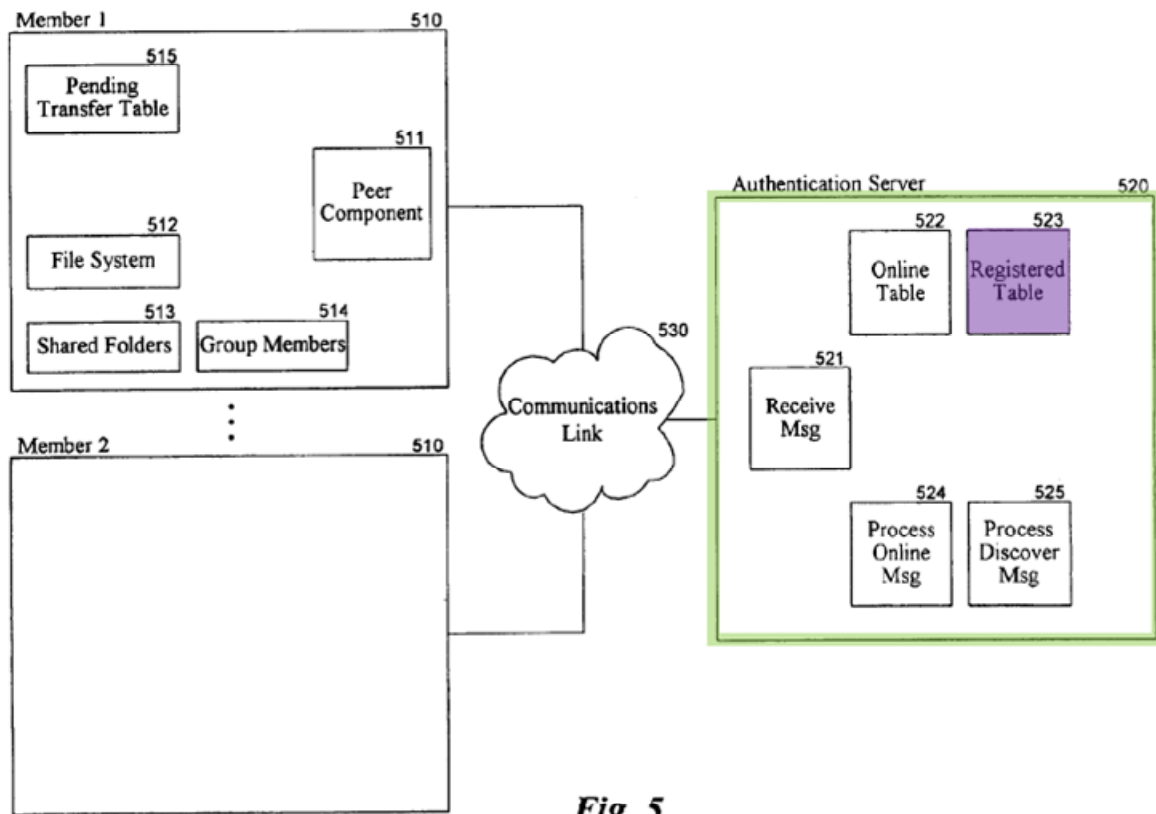


Fig. 5

Ex-1005, Fig. 5.

112. Aboulhoshn describes in Figure 6A the process “when a group owner invites a computer system to become a new member.” Ex-1005, 6:6-7.

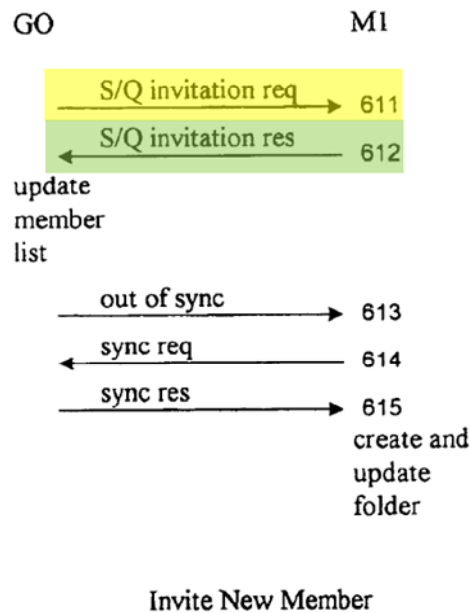


Fig. 6A

Ex-1005, Fig. 6A. “[G]roup owner [] sends an invitation request message 611 [yellow] to the invited computer system [M1]” and the “invited computer system sends an invitation response message 612 [green] to the group owner.” 6:13-15, 6:42-44. Once “the group owner **receives the invitation response message**, it **updates its member list for that group** accordingly.” *Id.*, 6:41-46. Therefore, a PHOSITA would have recognized that by registering a computer system to add it to its member list, Aboulhosen’s GO computer/server accepts registration of member devices, and this is done via the web-based system. *See* Section XII.A.1.b.

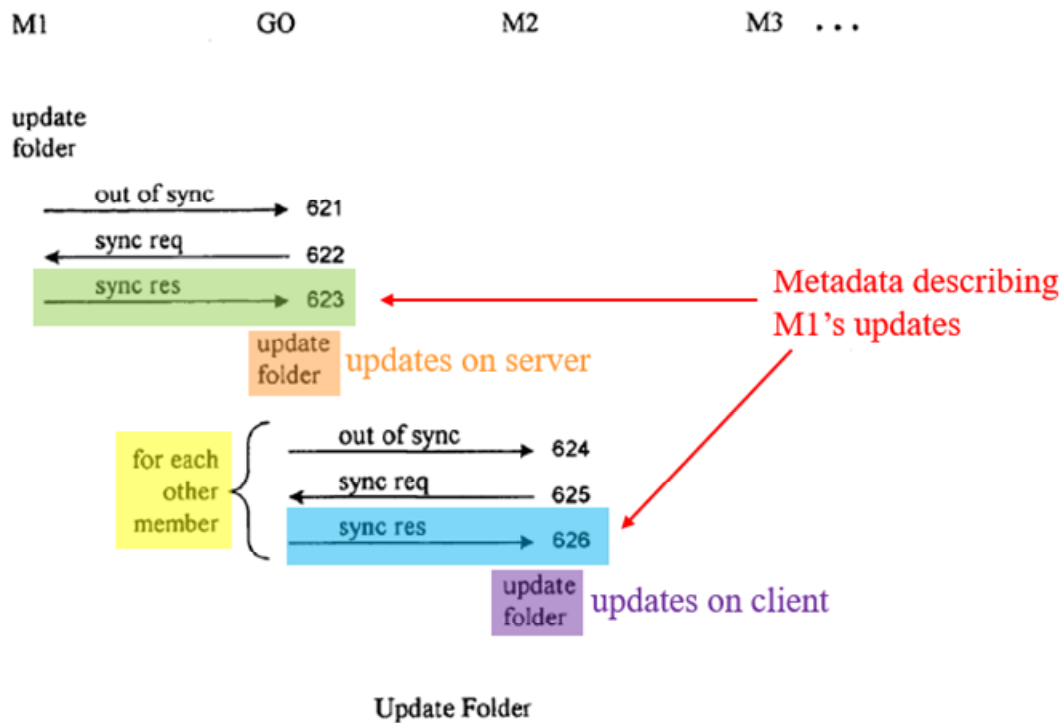
113. Therefore, in my opinion, Aboulhosen in view of Phillips discloses “**accepting registration of multiple devices** [e.g., Aboulhosen’s authentication

server receiving an invitation response message from each member device and creating an entry for a computer system in the registered table, and/or Aboulhosn's GO computer/server updating its member list to register a new member] **via the web-based system** [e.g., Aboulhosn's Internet, or Aboulhosn-Phillips's http/https and/or web browser].”

e. 1[D]

114. Limitation 1[D] requires “scanning each of the multiple devices by each of the individual software clients to inventory data on each of the multiple devices and create a meta-index of the files for the inventoried data.” In my opinion, Aboulhosn discloses this limitation.

115. It is my opinion that Aboulhosn discloses this limitation when **each member device adds or modifies files in the group folder**. Aboulhosn's Figure 6B illustrates “message flows when a member of the group updates the group folder...[including] **by adding**, deleting, or **modifying** a file of the shared folder.” Ex-1005, 6:60-62. As shown in Figure 6B below (which I have annotated to facilitate discussion), when M1 updates the group folder, it “sends an out-of-synchronization message 621 to the group owner,” who “sends a synchronization request message 622” back. *Id.*, 6:62-66. Then, M1 “responds with a synchronization response message 623 [green] that contains the **metadata describing the update**.” *Id.*, 6:66-7:1.

**Fig. 6B**

Ex-1005, Fig. 6B. Thus, when M1 adds or modifies files in the group folder, it creates a meta-index, *e.g.*, metadata describing the newly added or modified one or more files. In order to create such a meta-index, it is my opinion that a PHOSITA would have understood that M1 necessarily scans (*e.g.*, analyzes) its local version of the shared folder to inventory the newly added or modified one or more files, or would have found it obvious for M1 to do so. This is especially true when new files are added to an empty folder. In such situations, the only way to determine the state of the folder and the files contained therein is to scan (*e.g.*, analyzes) the folder. This is also true when subfolders are copied to the group folder. In order to

determine what files are contained in these subfolders, these subfolders must be scanned. In fact, scanning local file changes for file synchronization was well known and widely used in the field. For example, U.S. Patent Publication No. 2007/0283050 A1 discusses “[a] file system hierarchy [that] is built by a **scanning module** based on **recursive scanning** of the file system on the synchronization client computing device.” Ex-1040, Abstract. “**If a file state change is detected, the scanning module creates a synchronization report describing the file state change** and schedules transmission of the synchronization report to the synchronization server appliance device” (*id.*), just like Aboulhosn, where “when a member of the group **updates the group folder**... [t]he updating member responds [to the group owner] with a synchronization response message 623 that contains the **metadata describing the update**” (Ex-1005, 6:59-7:1).

116. I note that while Aboulhosn references “a file” (singular) in the above cited disclosure (*i.e.*, Ex-1005, 6:60-62 (“adding, deleting, or modifying **a file** of the shared folder”)), Aboulhosn discloses that a member device may update **multiple files** in the shared folder. *See, e.g.*, Ex-1005, 3:2-3 (“the member... may have added, deleted, or updated **files** of the group folder”). When that happens, the metadata describing the updates is sent to all other member devices.

117. Accordingly, it is my opinion that, when each of the multiple devices, such as M1, M2, and M3, adds or modifies one or more files, each of M1, M2, and

M3 scans the respective device to inventory the newly added or modified file(s) and creates a respective meta-index describing the updates.

118. In my opinion, Aboulhosn also discloses this limitation when the **GO computer/server comes online after being offline**. Aboulhosn illustrates in Figure 6E “message flows when a group owner comes online.” Ex-1005, 7:50-51.

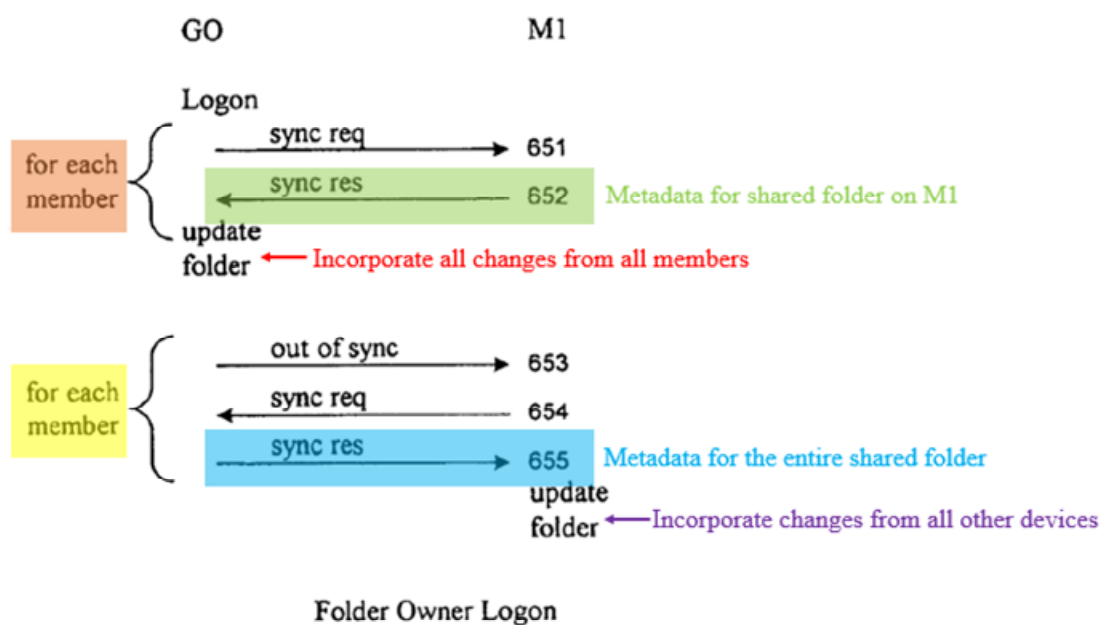


Fig. 6E

Ex-1005, Fig. 6E. As shown, “[w]hen a group owner becomes online, it sends a synchronization request message 651 to each member [orange]...[who] responds with a synchronization response message 652 [green] containing the **metadata for that shared folder**.” *Id.*, 7:51-57. Thus, when (1) GO is offline, (2) each of M1, M2, and M3 adds one or more files to the shared folder for the first time, and (3)

then GO becomes online, it is my opinion that a PHOSITA would have understood that each of M1, M2, and M3 necessarily scans (*e.g.*, analyzes) its local device for the newly added files to create a respective meta-index for the newly added one or more files (*i.e.*, “metadata for [the] shared folder”), or would have found it obvious for M1, M2, and M3 to do so.

119. In my opinion, a PHOSITA would have further understood that Aboulhosn’s scanning of file additions/modifications is performed by at least the peer component 511 (blue) on each member device, which “updates the metadata for each file in the group, creating new files as necessary” and “controls the sharing of files with other computer systems...including...sending updates to metadata, and so on.” Ex-1005, 8:51-52, 5:6-14.

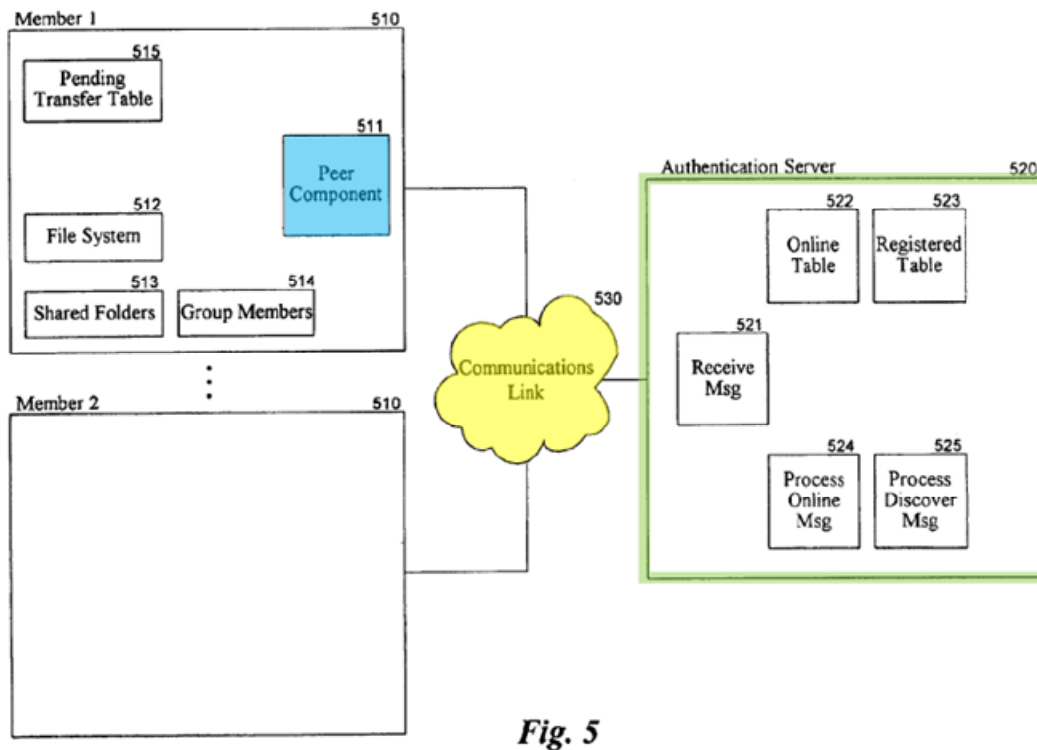


Fig. 5

Ex-1005, Fig. 5.

120. Therefore, in my opinion, Aboulhosn discloses “**scanning each of the multiple devices** [e.g., scanning each of the member devices M1, M2, and M3] **by each of the individual software clients** [e.g., at least the peer component on each member device] **to inventory data on each of the multiple devices** [e.g., inventory newly added/modified files on each member device (when member devices add/update files), or files added by each member device for the first time (when GO comes online after being offline)] **and create a meta-index of the files for the inventoried data** [e.g., metadata describing additions of/modifications to one or more files on each device (when all member devices add/update files), or

metadata for one or more files added by each member device for the first time (when the comes online after being offline)].”

f. 1[E]

121. Limitation 1[E] requires “providing by the individual software clients via the multiple devices individual meta-indices of the inventoried data for each of the multiple devices to the at least one server.”

122. As I explained in Section XII.A.1.e (1[D]) above, it is my opinion that Aboulhosn discloses that each of the member devices sends respective individual meta-indices of its respective inventoried data to Aboulhosn’s GO (the claimed “at least one server,” *see* Section XII.A.1.b) (1) when each of the member devices updates its respective group folder, and (2) when the GO comes online after being offline.

123. In my opinion, Aboulhosn also discloses that transmission of meta-indices is performed by the peer component 511 (blue) on each member device. *See* Ex-1005, 5:6-14 (“The peer component...controls the sharing of files with other computer systems...[and] **sending updates to metadata**, and so on”).

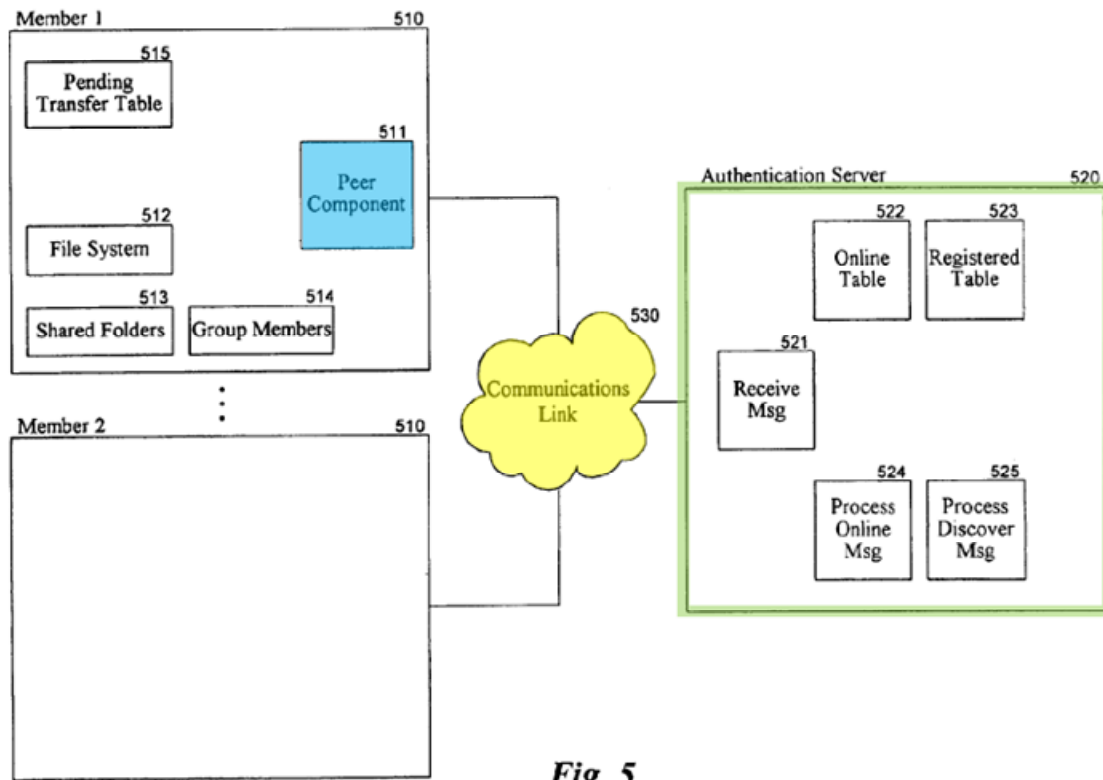


Fig. 5

Ex-1005, Fig. 5.

124. Therefore, in my opinion, Aboulhosn discloses “**providing by the individual software clients** [e.g., peer component 511 on each member device] **via the multiple devices** [e.g., member devices M1, M2, and M3] **individual meta-indices of the inventoried data for each of the multiple devices** [e.g., metadata describing file additions/modifications (when all member devices add/update files), or metadata for files added by each member device for the first time (when the GO comes online after being offline)] **to the at least one server** [e.g., Aboulhosn’s GO computer/server].”

g. 1[F]

125. Limitation 1[F] requires “integrating by the at least one server the individual meta-indices to create a single master meta-index.”

126. According to the '338 patent, the “master index of metadata” refers to metadata stored in a “database.” Ex-1001, 6:4-16 (“[T]he web service integrates all of the individual meta-indices for all of the user’s registered devices **into a database** S18. **This master index of metadata** thus becomes the singular file system that all devices use.”). In other words, a single master meta-index in the '338 patent is an aggregation of individual meta-indices from member devices.

127. In my opinion, Aboulhosn’s “metadata associated with the [entire] shared folder” (Ex-1005, 7:2-3) is a single master meta-index because the shared folder includes files from multiple member devices, and the metadata associated with the **shared folder** is an integration or aggregation of the metadata associated with the files from all member devices.

128. It is my opinion that Aboulhosn discloses the claimed integration and creation of a single master meta-index in at least two exemplary scenarios—(1) when each member device adds/modifies files in the shared folder, and (2) when the GO comes online after being offline.

129. **First**, when each of Aboulhosn’s member devices adds/modifies files in the shared folder, the GO receives “metadata describing the updates” via a

synchronization response message 623 (green) from M1. The GO then “updates [] the metadata associated with the shared folder [i.e., the singular file system] [orange].” Ex-1005, 7:2-3.

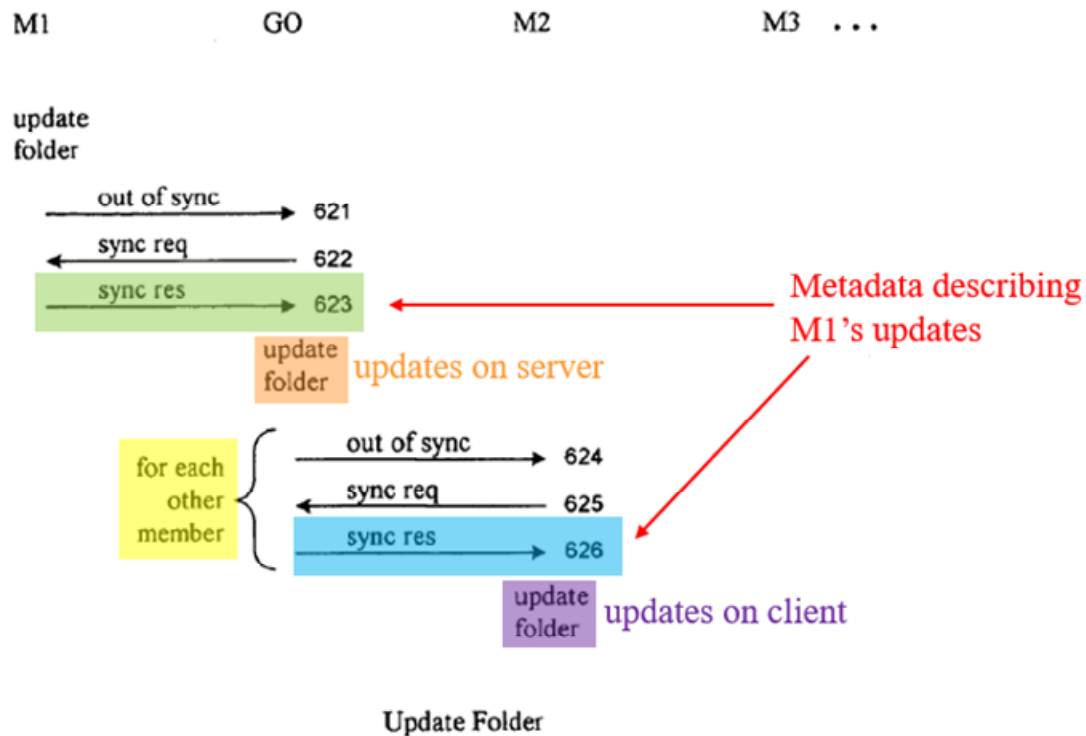


Fig. 6B

Ex-1005, Fig. 6B. A PHOSITA would have recognized that when each member device adds new files to the shared folder **for the first time**, the GO receives individual meta-indices (e.g., synchronization response message 623) from each member device and **creates** a single master meta-index (e.g., metadata associated with the shared folder) by combining or integrating individual meta-indices into the shared folder structure.

130. *Second*, when the GO comes online after being offline, “[e]ach member...[sends] a synchronization response message 652 [green] containing the metadata for that shared folder. The group owner then **updates its shared folder [red] to correspond to all the updates by any of the members** that occurred while the group owner was offline.” Ex-1005, 7:51-57. This process includes integrating the metadata received from each member device into the GO’s own metadata table for the entire shared folder, *i.e.*, the single master meta-index.

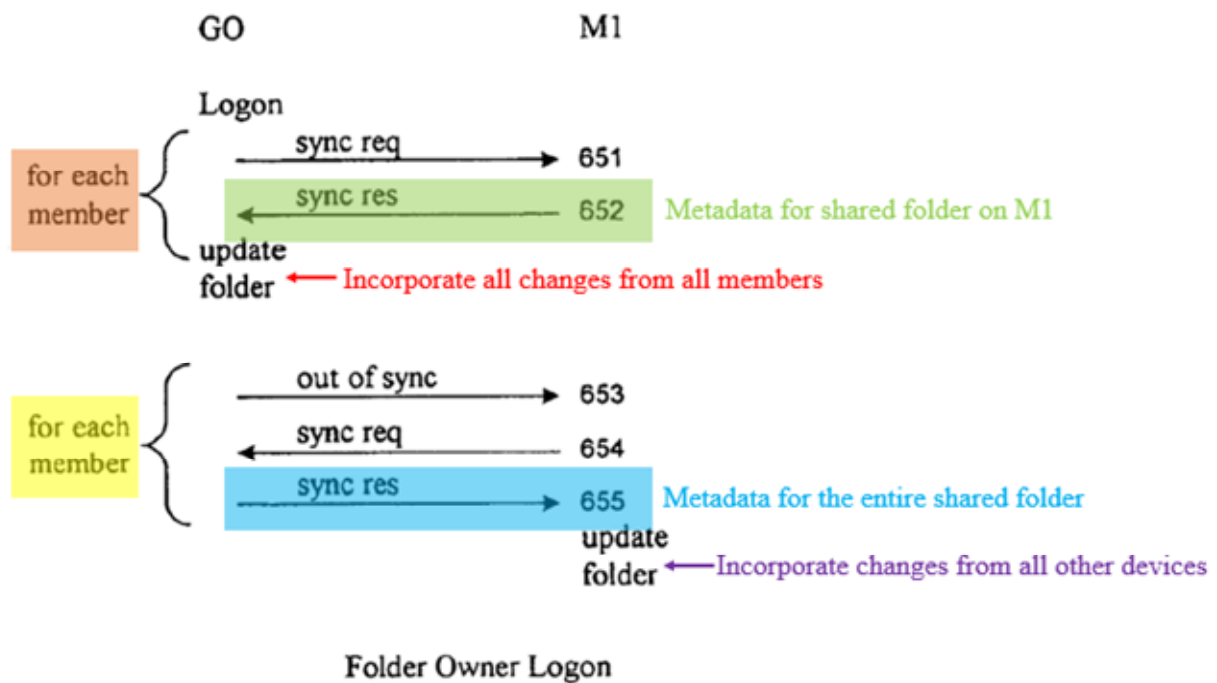


Fig. 6E

Ex-1005, Fig. 6E. This occurs at least where (i) the shared folder is empty when the GO goes offline, (ii) each member device adds new files to the shared folder,

and (iii) the GO then comes online. In this situation, the GO receives individual meta-indices from each member device and updates its shared folder to “create,” for the first time, a single master meta-index for the shared folder.

131. Therefore, in my opinion, Aboulhosn discloses “**integrating by the at least one server** [e.g., Aboulhosn’s GO as a server] **the individual meta-indices** [e.g., metadata describing file additions/modifications (when all member devices add/update files), or metadata for files added by each member device for the first time (when the comes online after being offline)] **to create a single master meta-index** [e.g., metadata associated with the entire shared folder].”

h. 1[G]

132. Limitation 1[G] requires “providing by the at least one server [i] the single master meta-index and [ii] meta-indices for each of the other multiple devices to each of the multiple devices via the individual software clients.” In other words, limitation 1[G] requires sending both (1) the single master meta-index, and (2) the “meta-indices for each of the other multiple devices” to each of the multiple devices. In my opinion, Aboulhosn discloses sending both.

133. *First*, Aboulhosn discloses “providing by the at least one server...[ii] **meta-indices for each of the other multiple devices** to each of the multiple devices.” As shown in Figure 6B below (to which I have added annotations to facilitate discussion), when M1 adds/modifies files in the shared folder, M1 sends

“**the metadata describing the update**” (synchronization response message 623, **green**) to the GO, which “**sends the metadata [describing the update] to the members (yellow)**” via a synchronization response message 626 [**blue**].” Ex-1005, 6:64-7:3, 7:4-11.

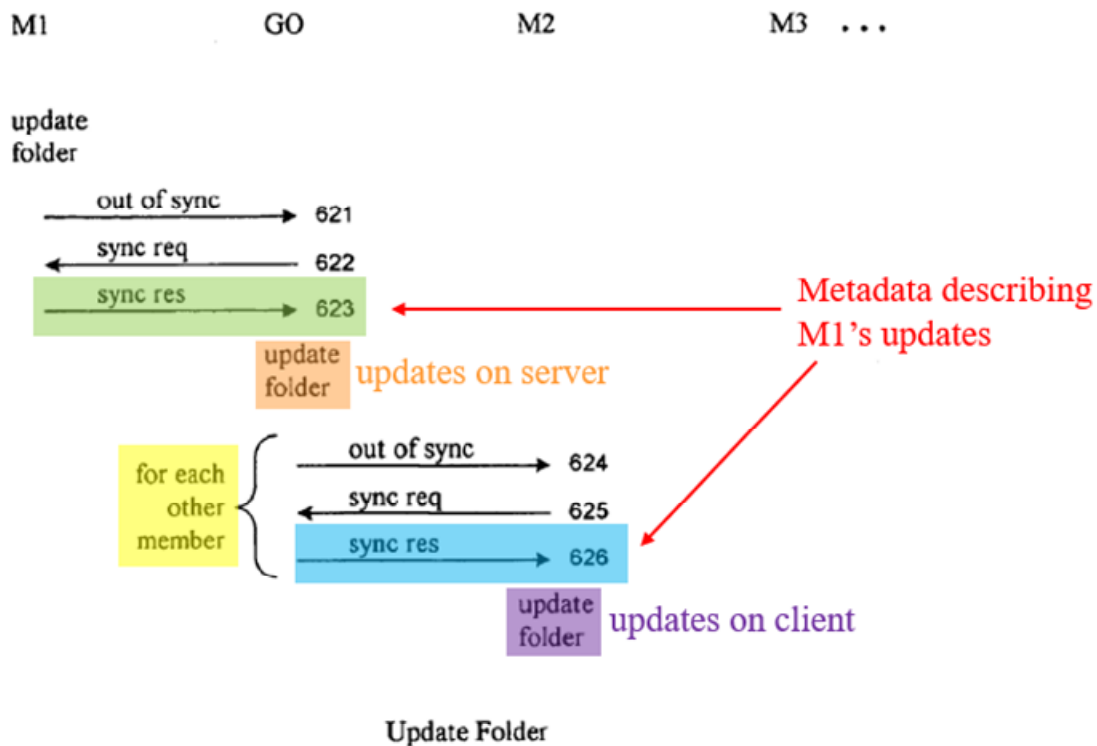


Fig. 6B

Ex-1005, Fig. 6B.

134. It is my opinion that a PHOSITA would have understood that “**the metadata**” sent by the GO via the synchronization response message 626 (**blue**) is **the** metadata describing the update, and is not metadata for the entire shared folder. Aboulhosn discloses several embodiments where synchronization response

messages are sent by the GO. When the message includes the metadata for the entire shared folder, Aboulhosn explicitly discloses this. *See, e.g.*, Ex-1005, 6:53-56 (when inviting a new member, “the group owner collects the **metadata for the shared files of the group** and sends that metadata to the new member”); 7:61-63 (when the group owner comes online after being offline, “[t]he group owner then sends **the current metadata for the shared folder** in a synchronization response message 655”). In contrast, for message 626 (blue) (Figure 6B above), Aboulhosn merely describes as containing “the metadata” (as opposed to “the metadata for the shared folder”), and in my opinion, a PHOSITA would have understood it to refer to metadata that the GO receives from individual member devices, *i.e.*, the metadata describing the update.

135. To the extent PO argues that Aboulhosn discloses with respect to Figure 10 that a “synchronization response message contains metadata for each file currently in the group” (Ex-1005, 8:49-51), this merely relates to “one embodiment” (*id.*, 8:49). A PHOSITA would have recognized that Aboulhosn’s synchronization response message does not always contain metadata for the entire shared folder. *See, e.g., id.*, 6:66-7:1 (“a synchronization response message 623 that contains the metadata describing the update”).

136. Moreover, in my opinion, a PHOSITA would have understood that, in the process depicted in Figure 6B, the GO provides the metadata from M1 to the

other member devices and that the GO sends out what it receives. It is also my opinion that a PHOSITA would have appreciated that there is no need for the GO to send the metadata for the entire shared folder to all other member devices when the only change to the shared folder is the metadata describing updates. In fact, sending the metadata that only concerns the update is sufficient for other member devices to sync, while sending the metadata for the entire folder would be highly inefficient and run counter to good network design practices.

137. Accordingly, when each member device adds/modifies files in the shared folder, the metadata describing the updates from each member device is provided by the GO (*i.e.*, the claimed at least one server) to each of the other member devices. *See also* Ex-1005, 2:43-44 (“All messages related to metadata updates are routed through the group owner.”).

138. *Second*, Aboulhosn also discloses “providing by the at least one server [i] the **single master meta-index**...to each of the multiple devices.” Aboulhosn discloses that when the GO comes online from being offline, it receives “**metadata for that shared folder**” (**green**) from each member device (**orange**) and “sends the **current metadata for the shared folder** in a synchronization response message 655 [**blue**] to each member [**yellow**].” Ex-1005, 7:61-63.

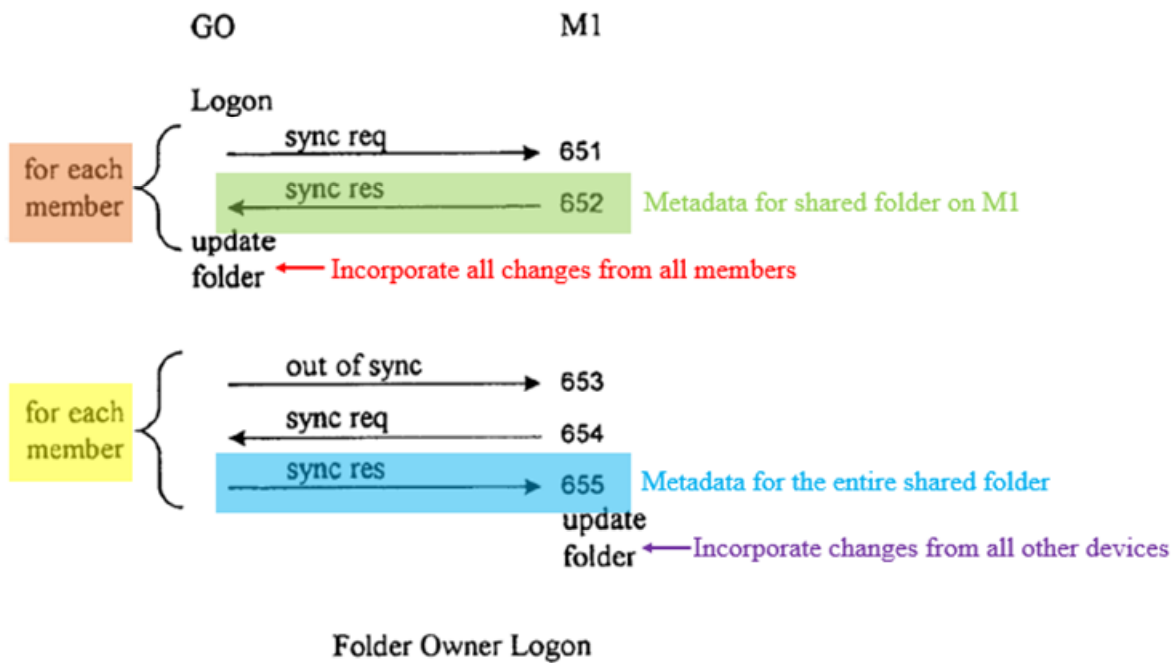


Fig. 6E

Ex-1005, Fig. 6E. In this scenario (as compared to Aboulhosn's Fig. 6B embodiment), it makes sense for the group owner to send the single master meta-index (as opposed to the individual meta-indices) as a matter of performance and simplicity. Specifically, the group owner will first collect all updates, aggregate them to have an updated state of the file system (*i.e.*, updated single master meta-index), and then send the updated single master meta-index to every member device. Thus, for n updates (e.g., assuming one update from each of n clients), the group owner only needs to send out the synchronization message n times (*i.e.*, one synchronization message for each member device). On the other hand, if the group owner were to simply follow the protocol for updating the individual meta-indices

(as disclosed in Aboulhosn's Figure 6B embodiment at Ex-1005, 6:61-7:15), then, for n updates (e.g., assuming one update from each of n clients), the group owner would have had to send $n*(n-1)$ synchronization message (*i.e.*, $n-1$ synchronization message for each member device). To see why this is so, consider that a small filesystem of 20 devices would require 380 messages, a medium system of 200 devices would require 39,800 messages, and a large network of 1000 devices would require 999,000 messages. Clearly, this provides substantial (and indeed prohibitive at scale) overheads and produces unnecessary additional traffic on the network.

139. As discussed above, Aboulhosn's peer component "controls the sharing of files with other computer systems," including "receiv[ing] a synchronization response message...[that] contains metadata for each file currently in the group." Ex-1005, 8:49-51. Therefore, in Aboulhosn, transmission of metadata is performed via the peer component (the claimed software clients).

140. Therefore, in my opinion, Aboulhosn discloses "**providing by the at least one server** [*e.g.*, Aboulhosn's GO as a server] **the single master meta-index** [*e.g.*, metadata for the entire shared folder when the GO comes online after being offline] **and meta-indices for each of the other multiple devices** [*e.g.*, metadata describing updates on each of the other member devices when each member device makes updates to a local file in the shared folder and the GO is already online] **to**

each of the multiple devices [e.g., each of the member devices M1, M2, and M3] **via the individual software clients** [e.g., peer component on each member device].”

i. 1[H]

141. Limitation 1[H] requires “integrating metadata from the meta-indices of each of the other multiple devices into a local file system of each of the multiple devices to generate virtual files stored in the same locations as local files of the local file system, the virtual files indistinguishable from the local files by the local file system at each of the multiple devices.” I will address different aspects of this limitation in turn below.

142. **Virtual files.** The ’338 patent “creates a **virtual representation** of data on all of the devices using **metadata indexing**. This virtual representation of data means that the files are not actually stored on all devices and thus, with the exception of a single physical location, they take up limited space (i.e. storage) on any additional device’s hard drive.” Ex-1001, 2:22-26. The claimed system purportedly “differentiate[s] data on a Local Device from data on a Remote Device” and “each client can create the **virtual representation of the remote data** within its local device file system.” *Id.*, 5:14-18. In my opinion, Aboulhosn discloses the exact same thing.

143. Aboulhosn discloses that “[t]he files in the folder of a member may be **actual files** stored at that member, which contains the content of the shared file, or **virtual files** identifying the actual file that is stored at another member.” Ex-1005, 2:5-8; 2:12-15 (“When a file is shared, the folder of the file owner contains the actual file and the folders of the other members of the group contain a virtual file corresponding to the shared file.”). Moreover, Aboulhosn’s “shared file is identified by **metadata stored in association with the virtual file** (e.g., as properties of the virtual file).” *Id.*, 2:8-9; 2:32-36 (“A virtual file may be a file that has the same name as the file being shared except for an identifying extension, may have no actual content (e.g., a file size of 0), and may have the metadata represented as attributes of the virtual file.”).

144. **Integrating metadata.** Aboulhosn discloses that when a member device adds/modifies files in the shared folder, “the [other] members receive from [the updating] member a notification...[and u]pon receiving...the member **creates a virtual file** in their folder for the group **using the file system of that member.**” *Id.*, 2:24-32. Accordingly, when each of the member devices adds/modifies files in the shared folder, each member creates within its local file system virtual files corresponding to newly added files by all the other member devices. *See also id.*, 6:56-57 (“When the new member receives the synchronization response message [that includes the metadata for the shared files of the group], it **creates the**

appropriate virtual files in the shared folder.”). In my opinion, a PHOSITA would have understood that such integration of metadata and creation of virtual files occurs when each member device updates its shared folder (**purple**) after receiving message 626 (**blue**)—the claimed “individual meta-indices.”

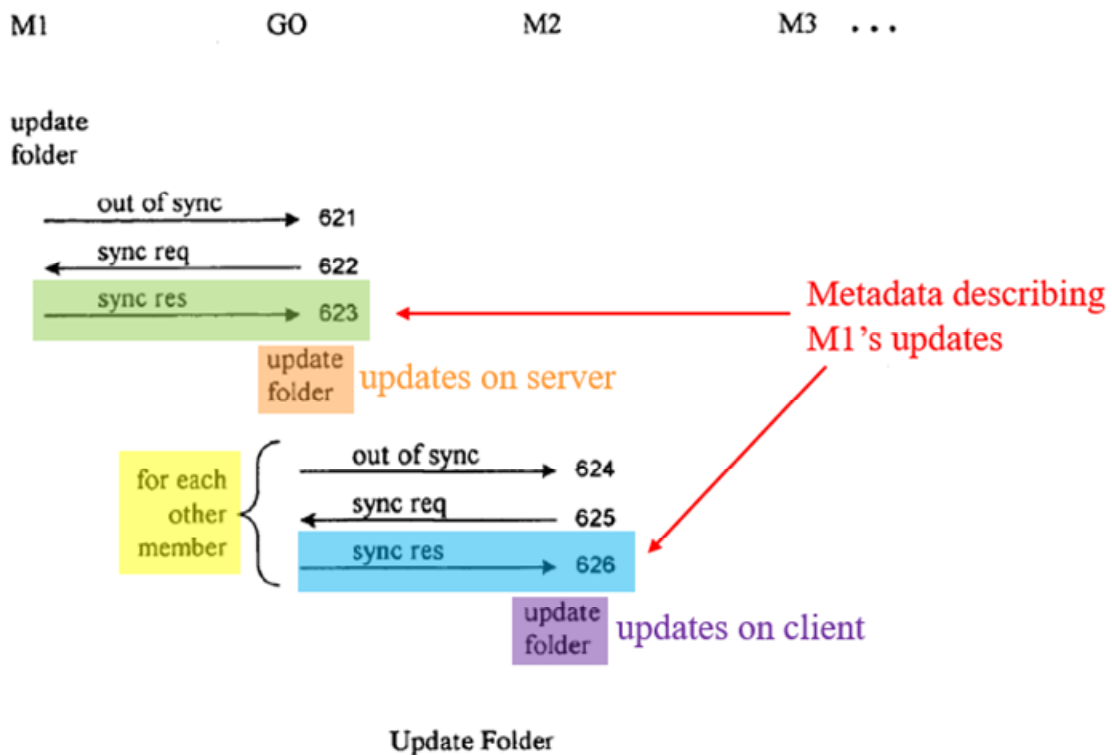


Fig. 6B

Ex-1005, Fig. 6B. In my opinion, such integration of metadata and creation of virtual files also occurs when each member device updates its shared folder (**purple**) after receiving message 655 (**blue**)—the claimed “single master meta-index.”

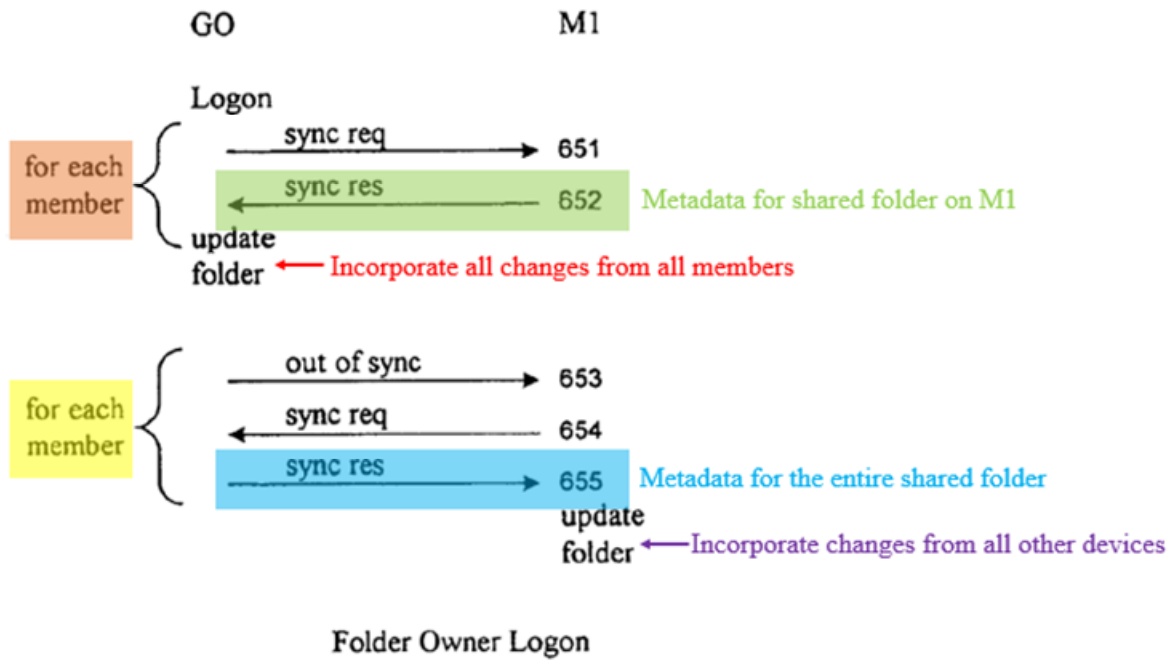


Fig. 6E

Ex-1005, Fig. 6E.

145. Aboulhosn illustrates in Figure 10 more details about how a shared folder can be updated on a member device. As shown below, when each member device receives a “synchronization response message [that] contains metadata for each file currently in the group,” the peer component “**creates new files** as necessary.” Ex-1005, 8:49-52.

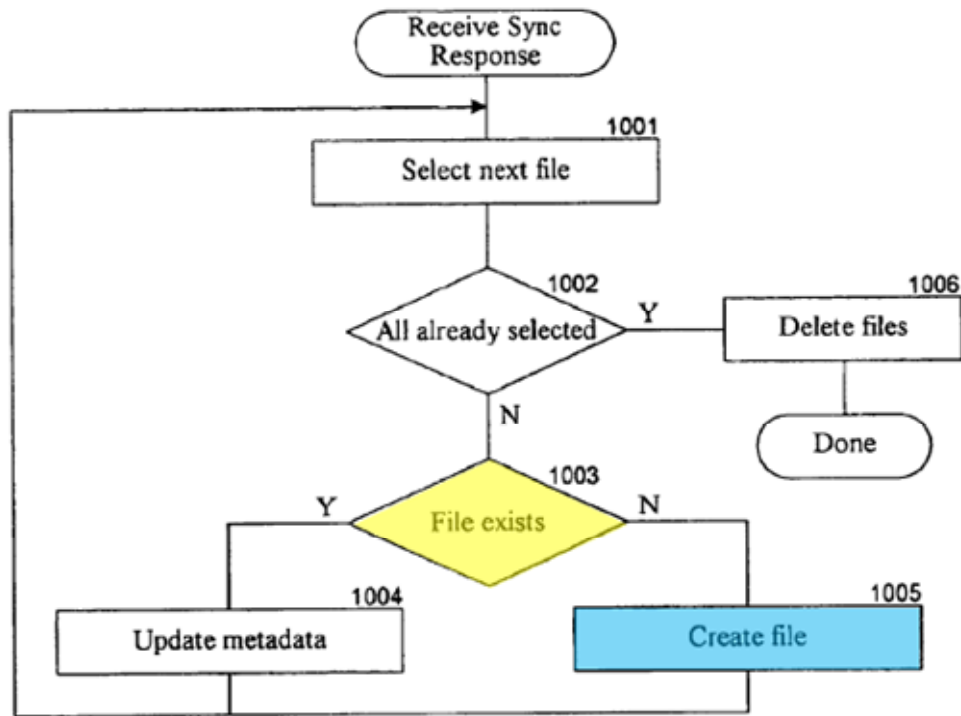


Fig. 10

Ex-1005, Fig. 10. For each file in the received metadata, if the file does not exist (block 1003, yellow), “the [peer] component **creates a new virtual file** [block 1005, blue] based on the metadata for that file.” *Id.*, 8:63-64; claim 1 (“**creating a virtual file using a file system** of the first computer system”). It is my opinion that a PHOSITA would have understood that this process also happens when the metadata for less than all files in the shared folder is sent to a member device.

146. **Same locations.** In Aboulhosn, virtual files are identified by the “file name extension of ‘vf’.” Ex-1005, 4:31-32.

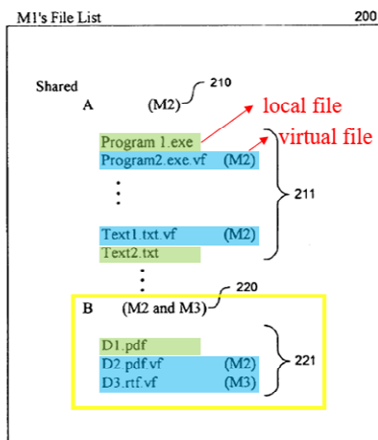


Fig. 2

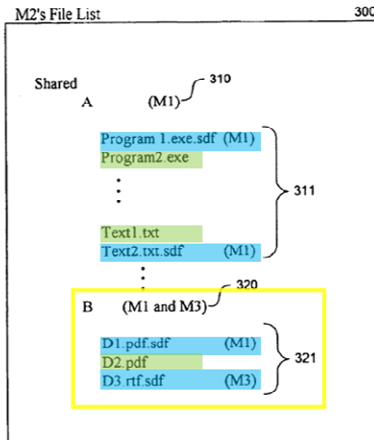


Fig. 3

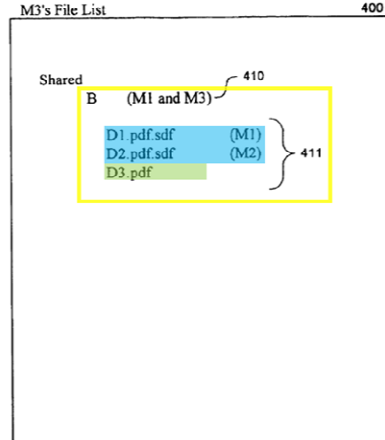


Fig. 4

Ex-1005, Figs. 2-4 (virtual files in blue and local files in green); 4:25-49. I note that while the virtual files in Aboulhosn's Figures 3-4 have the file name extension of "sdf," the relevant portions of the specification still describe "virtual files as indicated by the 'vf' extensions." Ex-1005, 4:54-55. For purposes of my declaration, "vf" and "sdf" are treated the same to both indicate virtual files.

147. Taking shared folder "B" (yellow) as an example, the shared folder B on each of M1, M2, and M3 includes both virtual file(s) (blue) and local file(s) (green). While Figures 2-4 above only illustrate M1, M2, and M3, each having a single local file and two virtual files in the shared folder B, it is my opinion that a PHOSITA would have understood that each member device may have any number of local files and any number of virtual files, and these local and virtual files are stored in the same folder, *i.e.*, the same location. *See also* Ex-1009 (448 Application), p.11, 9:27-30 ("while it is considered desirable to maintain virtual

files 401 in subdirectory 211 to more clearly segregate them from local files 301 in subdirectory 207, they could be maintained with local files 301 directly within subdirectory 207.”). To the extent Patent Owner argues that being “stored...in the same locations” excludes being stored in “a separate and segregated location for such remote data (such as a synchronization folder or an additional hard drive “mounted” on the device[]),” it is my opinion that Aboulhosn teaches the same thing. Like the ’338 patent, Aboulhosn also keeps the local files and virtual files in the same locations. As shown above, Aboulhosn’s local files and virtual files are stored in the same folder, e.g., shared folder B.

148. ***Indistinguishability.*** While Aboulhosn does not disclose that its virtual files are “indistinguishable from the local files by the local file system,” it is my opinion that Phillips does.

149. Phillips discloses that the “access to the designated files in a fashion which is **indistinguishable, by users of, and applications executing at**, the first client node, **with** access to one or more **files stored on a physical storage device that is locally present** at the first client node.” Ex-1006, claim 16. In my opinion, Phillips discloses both (1) visual indistinguishability and (2) transparent access.

150. ***Regarding visual indistinguishability***, the ’338 patent states that “[t]he virtual files **appear** in essentially all ways **the same** as a local file, making them **indistinguishable** unless such a distinction is enabled as a separate option.”

Ex-1001, 2:34-36. In my opinion, Phillips discloses the same. Phillips discloses that “the virtual storage device, and **its contents (i.e., all of the files, directories/folders, etc.** stored in the virtual storage device), **appears, to the client node user** and the application executing at the client node, **to be locally present.**” Ex-1006, 14:62-66.

151. The Applicant of the '338 patent stated during prosecution of the '260 *patent* (not this challenged patent) that Aboulhosn uses a different file extension (*i.e.*, “vf”) for virtual files and therefore “discriminate[s] and distinguish[es] between virtual files and local files.” Ex-1011, p.55. Phillips does not include a similar purported deficiency by disclosing that “[t]he [virtual] files ‘**appear,**’ *i.e.*, **entirely behave...as if locally present on the client nodes.**” Ex-1006, 9:36-39; 13:60-63 (“The Windows NT™ **operating system does not distinguish** between such storage devices from the perspective of **the graphical display to the user.**”). Thus, it is my opinion that a PHOSITA would have understood Phillips to teach that its virtual files appear indistinguishable to local files, including file extensions.

152. *Regarding indistinguishable access*, the '338 patent states that “[t]he software clients, in combination with the web service, **transparently** perform numerous commands to request, locate and retrieve a physical copy of the virtual file **without the knowledge of the requesting application or operating system.**” Ex-1001, 8:58-61. To do so, the '338 patent discloses that “the software client

intercepts the operation request,” retrieves the remote file “from the remote device to the local device,” and performs “the operation...on the file at the local device.”

Id., 6:49-55. In my opinion, Phillips discloses the same.

153. Phillips discloses that when “the application causes an access to another file (e.g., attempts to execute an application contained in another file or attempts to read, write, modify, etc. the data contained in another data file)...the application generates the appropriate request to the operating system to perform the appropriate file access operations... If the file is contained in the virtual storage device [*i.e.*, a virtual file], the client node software **intervenes and assists the operating system** in identifying the appropriate file and in **providing the data of the file to the operating system** to complete the access to the file (*i.e.*, read, write, modify, delete, etc.)[.] The client node software does this **transparently and automatically without requiring intervention by the user.**” *Id.*, 14:48-60; *see also*, 9:39-42 (“Any file or directory information transfer and integrity maintenance is performed **transparently to the user and applications...as if they were locally present.**”); claim 16. Thus, a PHOSITA would have understood that Phillips’s virtual file access is transparent to the user, the application, and the operating system. *See also, e.g.*, Ex-1006, 15:8-13 (“Thus, the actual providing of data to applications, useful display of the status or arrangement of files and directories/folder, etc. is performed by the operating system. The client node

software merely serves to locate and obtain valid copies of remotely stored or homed directory/folder information and file data”); *id.*, 27:7-10 (“In step S414, the client node attempts to perform the access operation on the file data or directory information, e.g., using the operating system or application which requested the access.”); *id.*, 26:45-49 (“Many of the attempted requested access operations are simply performed according to the normal operation of the operating system and/or application executing at the client node through which the access request was generated”).

154. Therefore, in my opinion, Aboulhosn in view of Phillips discloses **“integrating metadata from the meta-indices of each of the other multiple devices [e.g., metadata for modified or newly added files by each of the other member devices in Aboulhosn] into a local file system of each of the multiple devices to generate virtual files [e.g., Aboulhosn’s “virtual files”] stored in the same locations [e.g., in the same shared folder B] as local files [e.g., Aboulhosn’s “actual files”] of the local file system, the virtual files indistinguishable from the local files by the local file system [e.g., Aboulhosn’s virtual files as modified by Phillips to (1) appear visually indistinguishable to local files, and/or (2) be accessed transparently to the user, application, and operating system] at each of the multiple devices.”**

j. 1[I]

155. Limitation 1[I] requires “continually updating the single master meta-index on the at least one server and each of the multiple devices in response to changes to the data indexed thereon.”

156. As discussed above, Aboulhoshn illustrates in Figure 6B “the message flows when a member of the group **updates the group folder**...by adding, deleting, or modifying a file of the shared folder.” Ex-1005, 6:59-62. As shown below, when M1 updates the shared folder, “[t]he **group owner** upon receiving the metadata **updates [orange] to [sic] the metadata associated with the shared folder.**” *Id.*, 7:1-3.

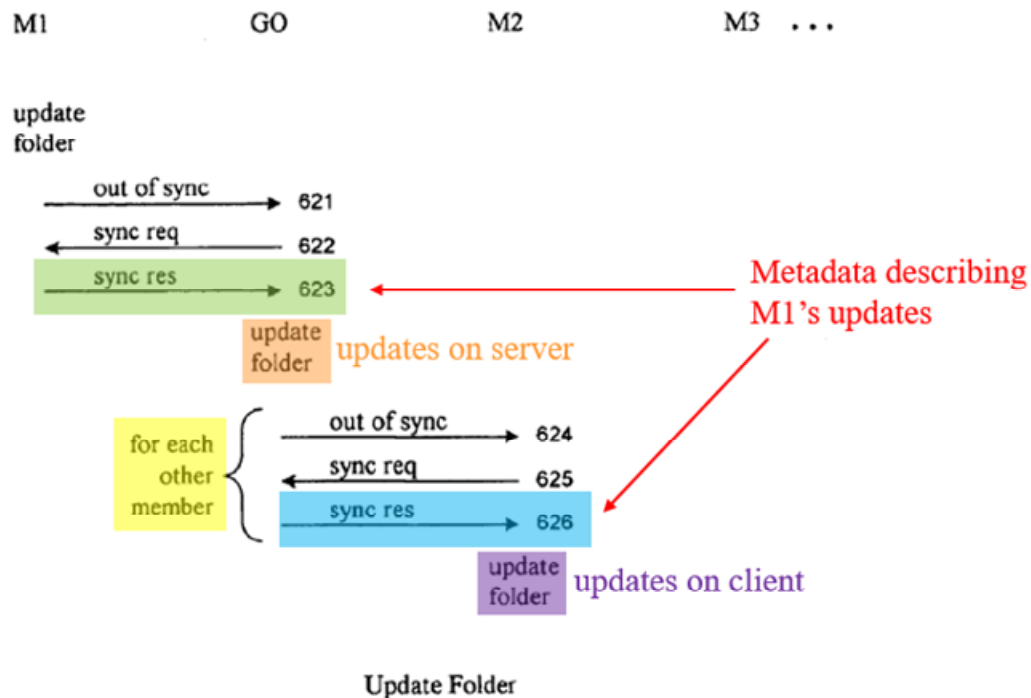


Fig. 6B

Ex-1005, Fig. 6B. Thereafter, “[u]pon receiving the metadata (blue), **the member updates [purple] its folder,**” including updating the local version of the metadata for the shared folder. *Id.*, 7:10-11.

157. As also shown in Figure 10 below, Aboulhosn discloses that, when data changes on a member device, each of the other member devices can “update[] the metadata for the existing file” at block 1004 (green) or “delete[] any virtual files of the shared folder that is not represented by the metadata” at block 1006 (red). *Id.*, 8:61, 8:65-57.

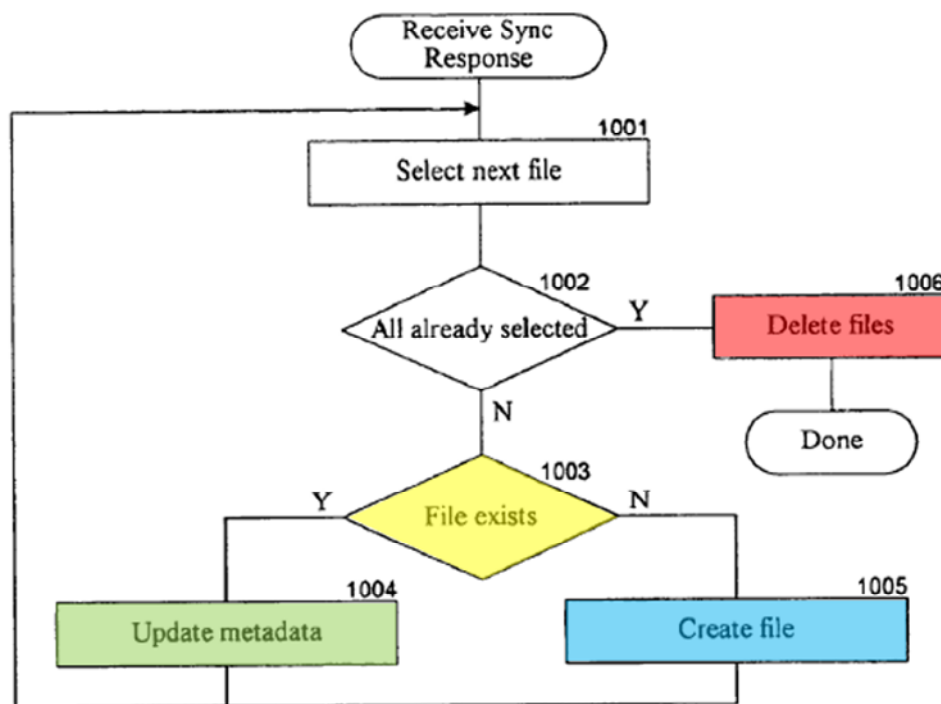


Fig. 10

Ex-1005, Fig. 10.

158. In my opinion, a PHOSITA would also have understood that the member device that initiates the data change to an existing file of the shared folder necessarily updates the metadata for such file and thereby updates its local metadata for the shared folder. Therefore, Aboulhosn's single master meta-index (e.g., the metadata for the shared folder), which is maintained by both the GO and the member devices, is updated in response to data changes to files already in the shared folder. *See also*, Ex-1005, claim 1 ("receiving...metadata for the file and **updating the metadata** stored in association with the virtual file based on the received metadata"); Figs. 6B, 6D, 6E.

159. Since the above synchronization process occurs whenever there is a data change, it is my opinion that it is performed continually.

160. Therefore, in my opinion, Aboulhosn discloses "**continually updating the single master meta-index** [e.g., metadata for the shared folder is updated whenever there is a change] **on the at least one server** [e.g., Aboulhosn's GO as a server] **and each of the multiple devices** [e.g., each of the member devices M1, M2, and M3] **in response to changes to the data indexed thereon** [e.g., modifying, adding, or deleting file(s) in the shared folder]."

k. 1[J]

161. Limitation 1[J] requires "wherein the individual software clients facilitate storage of the data within each of the multiple devices in accordance with

the single meta-data index by modifying file systems of each of the multiple devices to include virtual files for data from the single meta-data index that is not local to a multiple device.”

162. For reasons I have explained in Section XII.A.1.i (1[H]), Aboulhosn discloses that “the individual software clients facilitate storage of the data within each of the multiple devices in accordance with the single meta-data index...to include virtual files for data from the single meta-data index that is not local to a multiple device” as required by this limitation.

163. In my opinion, Aboulhosn also discloses “**modifying file systems** of each of the multiple devices to include virtual files,” as required by this limitation. Aboulhosn’s system includes a file system 512 (red), which “may be a conventional file system provided by an operating system.” Ex-1005, 5:14-15; *see also id.*, claim 1 (“creating a virtual file using a **file system** of the first computer system”).

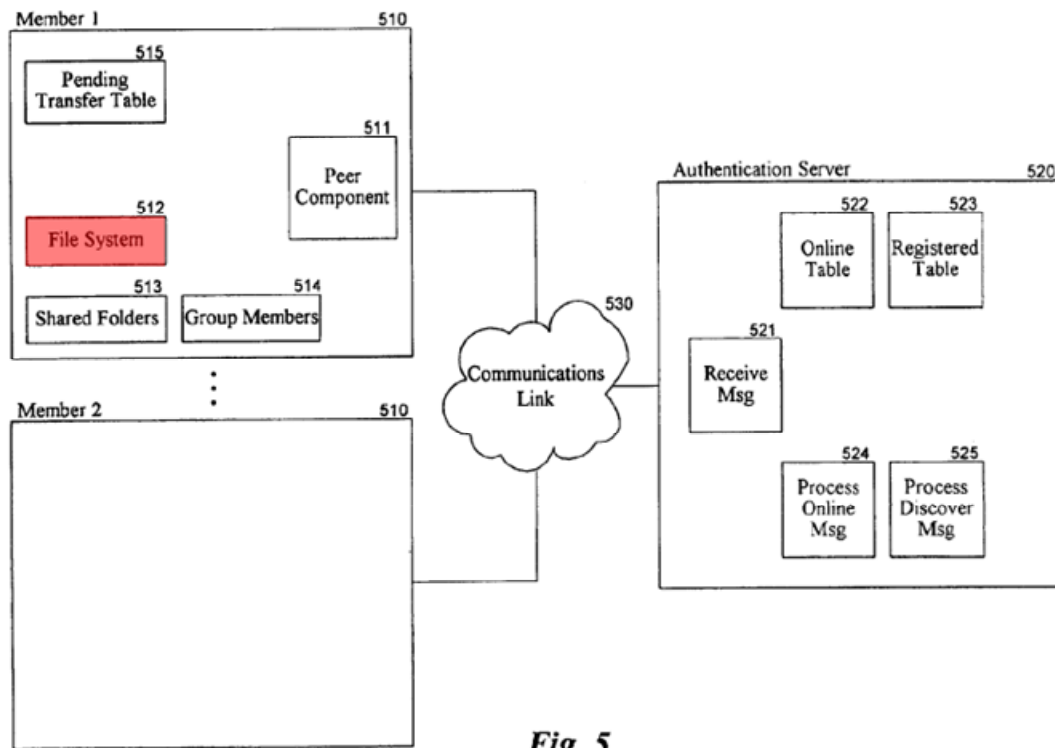
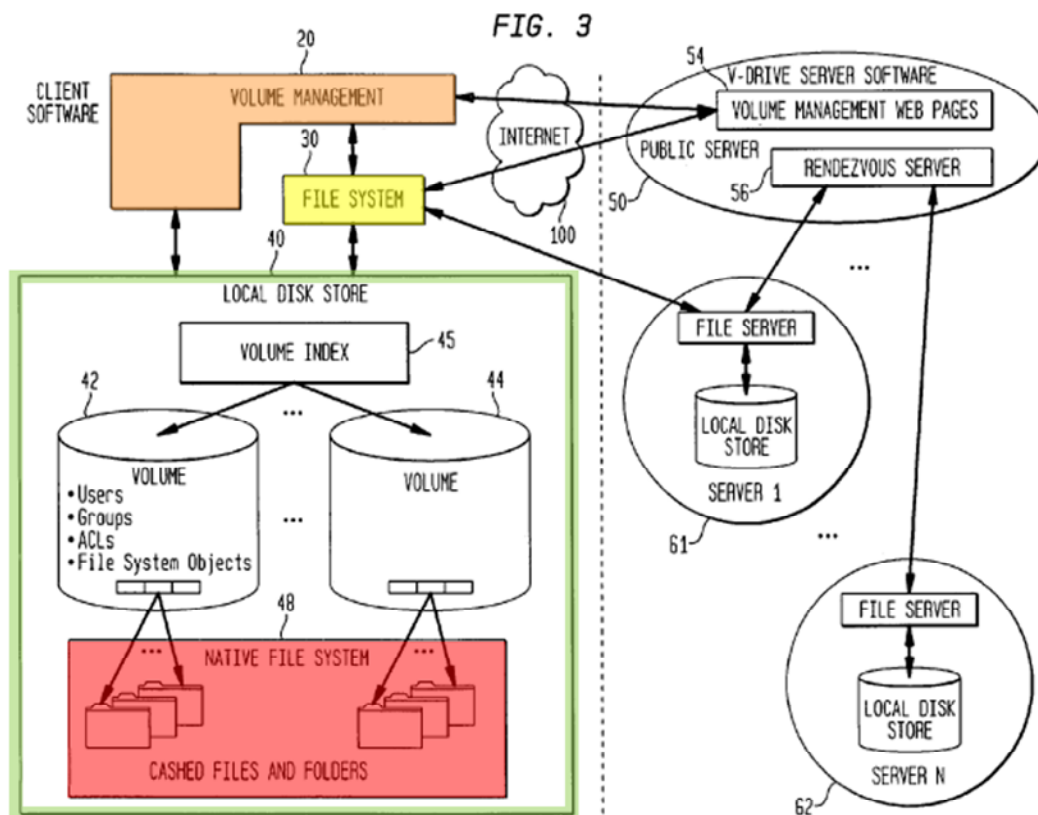


Fig. 5

Ex-1005, Fig. 5. Aboulhosn also discloses that “a virtual file component receives a notification from the file system.” *Id.*, claim 3. In my opinion, a PHOSITA would have understood that Aboulhosn’s virtual file system—including at least the implementation of shared folders and the sending of notifications to Aboulhosn’s virtual file component—are not part of a conventional operating system and are accomplished through at least the peer component 511 “downloaded from [Aboulhosn’s] authentication server.” *Id.*, 5:7-8; claim 39 (“a virtual file component that creates a virtual file”). Therefore, a PHOSITA would have understood that to implement Aboulhosn’s system, including creation of virtual

files and providing notifications when virtual files are accessed, the conventional file system of each member device is necessarily modified.

164. To the extent PO argues that Aboulhosn does not disclose modifying a file system, it is my opinion that Phillips does. As shown in Figure 3 below, Phillips discloses “three client node software elements, namely, the volume management [orange], file system [yellow] and disk subsystem [green]. These software elements are designed to **integrate with a conventional operating system/native file system 48 [red]** which may be sold with the client node.” Ex-1006, 13:7-11.



Ex-1006, Fig. 3. Phillips elaborates that such integration is OS-specific: “For example, Microsoft™ specifies an API for integrating software **affecting the manner by which files are identified and retrieved by other applications and programs** executed within the Windows NT™ operating system.” Ex-1006, 13:16-20. How files are identified and retrieved is the responsibility of the file system, and, thus, software that would “affect” this process necessarily requires modifying the file system. Therefore, Phillips discloses implementing its system, including creating virtual files, by modifying the file system, such as by “integrat[ing software elements] with the...native file system.” *Id.*

165. Therefore, in my opinion, Aboulhosn in view of Phillips discloses “**wherein the individual software clients** [*e.g.*, Aboulhosn’s peer component 511 on each member device] **facilitate storage of the data within each of the multiple devices in accordance with the single meta-data index** [*e.g.*, metadata for the entire shared folder] **by modifying file systems** [*e.g.*, modifying Aboulhosn’s existing file systems to communicate with the installed peer components, or modifying Aboulhosn’s existing file systems to integrate with the installed peer components as taught by Phillips] **of each of the multiple devices to include virtual files** [*e.g.*, Aboulhosn’s “virtual files”] **for data from the single meta-data index that is not local to a multiple device.**”

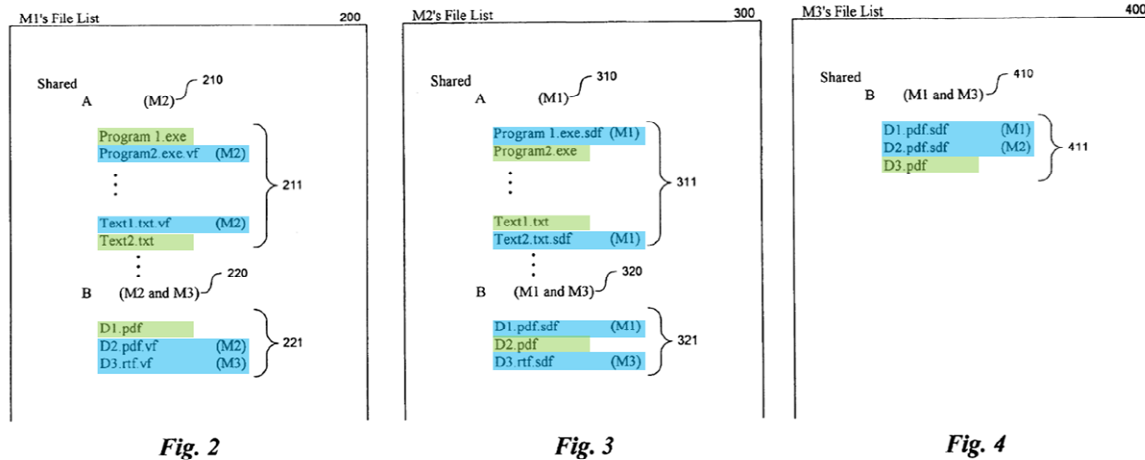
2. Motivation to Combine Aboulhosn and Phillips

166. In my opinion, Aboulhosn and Phillips are **analogous art**, at least because they are in the **same field of endeavor** as the '338 patent, *i.e.*, file sharing and synchronization. The '338 patent and both references are in CPC class G06F and U.S. class 707 (I understand that CPC class and U.S. class are used by the US Patent Office to classify patents based on technology areas, and such classification information appears on the face of patents and patent publications). All disclose a client component residing on a client device to facilitate file sharing and synchronization. Ex-1001, claim 1; Ex-1005, 5:6-14; Ex-1006, 11:16-67. Moreover, all relate to the same problem of needing to reduce storage space occupied by shared files on a client device and endeavor to solve that problem in a similar manner by generating virtual files for files that are shared between multiple devices. Ex-1001, Abstract; Ex-1005, 2:32-36; Ex-1006, 14:62-66. When a client device accesses a virtual file, all three references disclose transmitting the actual file from the remote device to the client device to allow operations on the file. Ex-1001, 8:57-61; Ex-1005, 2:15-18; Ex-1006, 14:38-42, 28:21-31. Thus, it is my opinion that, at a minimum, Aboulhosn and Phillips are directly focused on addressing the same problem identified in the '338 patent regarding files occupying too much space on a local device by keeping a copy of all shared files on each device. Ex-1001, 1:40-46; Ex-1005, 2:20-23; Ex-1006, 2:60-63, 11:59-61.

Like the '338 patent, Aboulhosn and Phillips address this problem by generating virtual files. Ex-1001, Abstract; Ex-1005, 2:32-36; Ex-1006, 14:62-66.

167. **Indistinguishability**. In my opinion, a PHOSITA would have been motivated to incorporate Phillips's virtual file features, such as visual indistinguishability and transparent access, into Aboulhosn's system.

168. The '338 patent recites “the virtual files **indistinguishable** from the local files by the local file system.” Ex-1001, claim 1. The '260 patent, which is a continuation of the '338 patent, recites a similar limitation—“make local files and virtual files **appear indistinguishable to the singular file system**.” Ex-1010, claim 1. During prosecution of the '260 patent (not the '338 patent), the Applicant distinguished Aboulhosn from the '260 patent, because “the file system in Aboulhosn uses the **file name extension ‘vf’ to discriminate and distinguish between virtual files and local files**.” Ex-1011, p.55. As shown in Figures 2-4 below, Aboulhosn's virtual files (**blue**) also appear different from the local files (**green**) in shared folder B due to the virtual file extensions.



Ex-1005, Figs. 2-4.

169. In my opinion, a PHOSITA looking to provide a smoother and more seamless user experience with Aboulhosen's system would have been motivated to incorporate Phillips's **visual indistinguishability feature** (*i.e.*, displaying virtual files to the user in the same way as local files) to Aboulhosen's system. In this way, "the client user...is [not] aware of the actual location or home of the files as the integration is **perfectly transparent and seamless.**" Ex-1006, 15:4-7. By presenting virtual files as indistinguishable to local files, users accessing a file can defer to software to automatically perform these steps. *See* Ex-1006, 14:58-60 ("The client node software does this **transparently and automatically without requiring intervention by the user.**"). This provides the perception that all the shared files are synced/stored locally while the local disk drive need only provide minimal space for the remote/virtual files. For users that are not familiar with

Aboulhosn's file extension naming conventions, removing Aboulhosn's special virtual file extension "vf" also avoids confusion.

170. In my opinion, a PHOSITA would also have been motivated to incorporate Phillips's **transparent access feature** for the virtual files, *i.e.*, accessing files transparently to the user, application, and operating system. When accessing a remote file, both Aboulhosn and Phillips disclose retrieving the remote file to the local device to allow operations. Ex-1005, 2:15-18; Ex-1006, 14:38-42, 28:21-31. Because Aboulhosn does not provide details on how such file retrieval is implemented, it is my opinion that a PHOSITA implementing Aboulhosn's system would have been motivated to look to other references, like Phillips, for details of how to access remote files. For example, Phillips teaches that "the application generat[ing] the appropriate request to the operating system to perform the appropriate file access operations" and "the client node software intervenes and assists the operating system in identifying the appropriate file and in providing the data of the file to the operating system to complete the access to the file (*i.e.*, read, write, modify, delete, etc.)." Ex-1006, 14:52-58. Phillips goes further, teaching that "[t]he client node software does this **transparently and automatically without requiring intervention by the user,**" and that "[i]n essence, neither the client user nor the application executing at the client node is aware of the actual location or home of the files as the integration is **perfectly transparent and**

seamless.” Ex-1006, 14:58-60, 15:4-7. Therefore, Phillips’s implementation of remote file access also provides the benefit of a better, smoother, more efficient, and more seamless user experience. For example, when a user wishes to access a file that is stored remotely, Phillips automatically retrieves the file content on behalf of the user. In this way, the user needs not be concerned about whether the file is local or not.

171. In my opinion, a PHOSITA would have had a **reasonable expectation of success** in incorporating Phillips’s visual indistinguishability feature and transparent access feature into Aboulhosn’s system. Both systems employ client software to facilitate virtual file creation and enable remote file access from a client device. Ex-1005, 5:6-14, 7:16-24; Ex-1006, 11:22-67, 14:54-58. In order to display virtual files and local files in the same manner as taught by Phillips, it is my opinion that a PHOSITA would have understood that the modifications to Aboulhosn’s system merely include removing the “vf” extensions for virtual files, *e.g.*, removing “.vf” from “D2.pdf.vf.” *See id.*, Fig. 2.

Aboulhosn’s peer component need not rely on the “vf” extensions (to the extent it did) to determine whether a file is local or remote because Aboulhosn’s “metadata may identify...the file owner.” Ex-1005, 2:28-30. In my opinion, a PHOSITA would have understood that changing file extensions, including removing special file extensions, could be performed using simple and well-known techniques

employed in file system naming implementations and that this change would have resulted in a working system. With respect to incorporating Phillips's transparent access feature, since Aboulhosn does not provide details on how to implement the file system for access, it is my opinion that a PHOSITA would have looked to Phillips and would have expected the combination to result in a working system for the above reasons.

172. **Network communications.** To the extent Aboulhosn were found to not disclose the claimed "web-based system" (it does), a PHOSITA would have been motivated to incorporate Phillips's use of http/https protocols for network communications into dispersed network systems, such as Aboulhosn's system. Aboulhosn's computer systems and servers communicate "via...signal[s] on a communications link [yellow]...such as the **Internet**, a local area network, a wide area network, or a point-to-point dial-up connection." Ex-1005, 5:57-61.

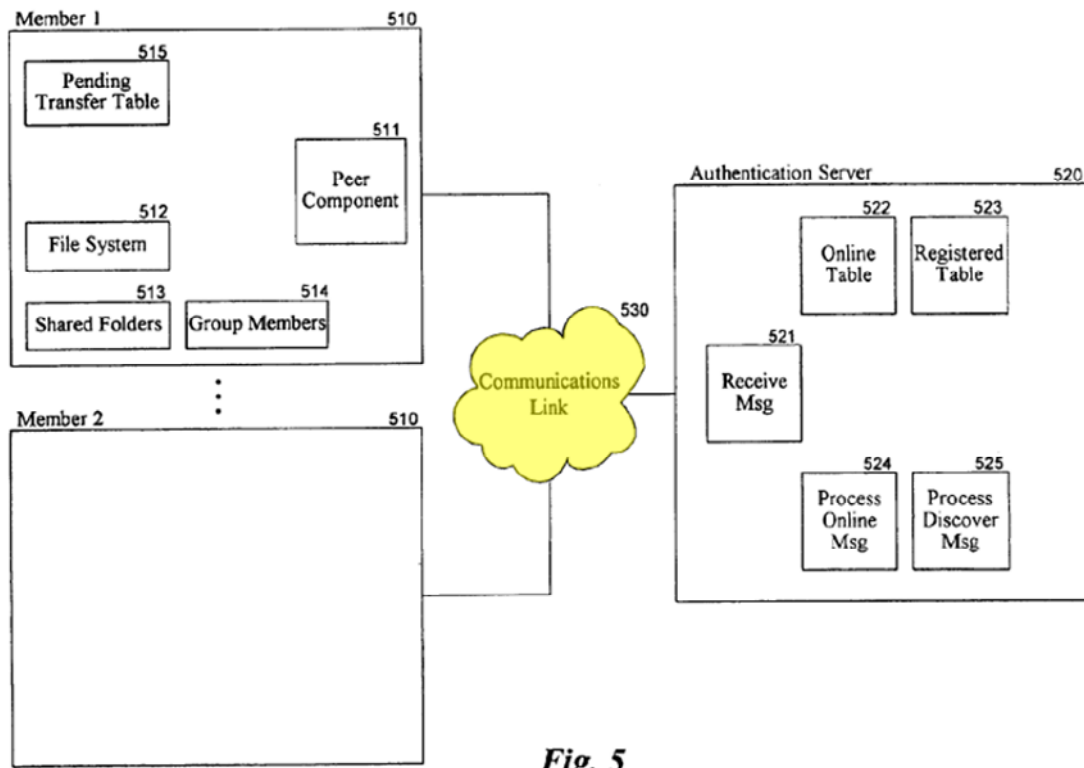


Fig. 5

Ex-1005, Fig. 5. In my opinion, because Aboulhosn discloses using an Internet-based communications link as well as use of FTP protocol via the Internet (Ex-1005, 1:24-25), a PHOSITA would have been motivated to use, and seek out a reference disclosing, common web-based Internet protocols such as http/https used in Phillips's web-based file-sharing and synchronization system. *See* Ex-1006, 8:9-14 (“Communication is achieved in the **Internet** using a hierarchy of protocols, including...**the hypertext transfer protocol (http) and the hypertext transfer protocol over secured sockets layer (https).**”).

173. In fact, http/https was the most popular protocol used for Internet communications at the time of the '338 patent. HTTP/1.1 was first officially

released in 1997. Ex-1014, p.1. HTTPS, also referred to as HTTP over TLS, was formally specified in May 2000. Ex-1015, p.1. Since their releases, http/https have become one of the most commonly used protocols. *See, e.g.*, Ex-1016, p.8 (“One of the most commonly used services on the Internet is the World Wide Web (WWW). The application protocol that makes the web work is Hypertext Transfer Protocol or HTTP.”); Ex-1017, p.2 (“The majority of widely known protocols used on the internet are layered on top of TCP. These include...the Hypertext Transfer Protocol (HTTP), used for accessing websites”). Its dominance in web browsing, content delivery, and other internet services would have made http/https a compelling choice for a PHOSITA designing an internet-based file sharing system. In fact, it is my opinion that a PHOSITA would have looked to http/https protocols specifically given that it is readily compatible with a variety of clients (e.g., smartphones, laptops, desktops, etc.) and thus avoids requiring specialized software for implementing a custom transfer protocol. *See, e.g.*, Ex-1016, p.8 (“HTTP is the protocol that web browsers and web servers use to communicate with each other over the Internet.”); Ex-1024; Ex-1025.

174. To the extent the claims require a web browser, it is my opinion that a PHOSITA would have been motivated to incorporate Phillips’s web browser into Aboulhosn’s Internet-based system. As I explained in Sections XII.A.1.b-XII.A.1.d, Aboulhosn discloses receiving user information, installing an individual

software client, and accepting registration of multiple devices, all via its Internet-based system. A PHOSITA implementing Aboulhosen's system would have been motivated to incorporate Phillips's web browser as a specific user interface to receive user information, to download the software client, and to enable GO's management functionalities, including accepting device registration.

175. In fact, at the time, web browsers were commonly used for these purposes and were one of the most common types of user interface in Internet-based communications. For example, FolderShare, a file sharing and synchronization software before the '338 patent, uses the web browser to manage accounts, invite others, and download software for installation:



Getting Started Guide

How it Works

FolderShare is a service that allows you to securely keep files synchronized between your devices, share files with friends or colleagues, and remotely download your files from any web browser. FolderShare consists of two components - My FolderShare and the FolderShare Satellite.

My FolderShare: Configure and manage your account [from here](#).

- Go to www.foldershare.com from any web browser and login to access your "My FolderShare" page.
- From here you can do any of the following:
 - Setup devices to sync or share with
 - Invite others (via email address) to share your files
 - Manage your account - upgrade your subscription, change your email address or password
 - Get help
 - Download the latest software

Ex-1041 (accessible at

<https://web.archive.org/web/20060313171107/https://www.foldershare.com/download/files/FolderShareGuide.pdf>)

176. Moreover, Aboulhosn specifically identifies a problem with distributed file systems that is directly solved by web-based systems: “Distributed systems reduce the need for centralized file storage, but they may suffer from one or more disadvantages **such as an unfamiliar user interface**, a need for substantial amount of user training, lack of up-to-date information on files currently available for retrieval, and so forth.” Ex-1005, 1:25-31. To solve such “unfamiliar user interface” problem identified in Aboulhosn, a PHOSITA would have been motivated to incorporate Phillips’s web browser, which is the most common and standardized interface for, as mentioned above, providing services such as account creation, software client download, and device management.

177. In my opinion, a PHOSITA would have had a **reasonable expectation of success** in incorporating Phillips’s http/https protocol and web browser in Aboulhosn’s system. As I mentioned above, like FTP disclosed in Aboulhosn, http/https is one of the most widely used protocols for Internet communications and was almost certainly used for Aboulhosn’s Internet communications. To the extent it was not, it would have been simple for a PHOSITA to utilize Phillips’s http/https protocol for Aboulhosn communications

link. *See, e.g.*, Ex-1005, 3:15-18 (“A group owner may **invite other computer systems to join** the group as members. In one embodiment, a computer system is eligible to be a member of a group after the computer system has registered with the authentication server.”); Ex-1006, 17:21-27 (“The activation of the **join process** results in the transmission of a message from the client node to the appropriate remote file server node, e.g., communicating the user name and identifier for the virtual storage device to the remote file server node. Illustratively, this is achieved using the so-called **https protocol**.”). In my opinion, it would also have been simple and routine to implement a web browser in Aboulhosn’s system for account/device registration and file sharing. *See* Ex-1041; Ex-1042, 17-21 (“the server (such as a web server 440 in FIG. 4) provides an interface through which a user can view and edit files from a plurality of devices (step 210). In one embodiment, the interface is a web interface displayed in a user's browser.”).

178. **Receiving user information.** To the extent Aboulhosn were found to not disclose receiving user information (it does), it is my opinion that a PHOSITA would have been motivated to incorporate Phillips’s creation of accounts into Aboulhosn’s system. As I explained in Section XII.A.1.b, Aboulhosn discloses creating a group of authorized users to access a shared folder. Ex-1005, claim 39. In my opinion, a PHOSITA would have been motivated to look to other references, like Phillips, for details, including receiving user information to create accounts.

Implementation of such a process is simple and routine and would have resulted in a working system.

179. **Modifying file systems.** To the extent Aboulhosn were found to not disclose modifying file systems (it does), it is my opinion that a PHOSITA would have been motivated to incorporate Phillips's disclosure of modifying file systems into Aboulhosn's system. As I explained in Section XII.A.1.k, Aboulhosn discloses a peer component that "controls the sharing of files with other computer systems." Ex-1005, 5:6-9. In my opinion, a PHOSITA looking to deploy the peer component on multiple devices would have been motivated to look to other references, such as Phillips (which also discloses virtual files), for details of how a client software interacts with existing file systems. Given that Phillips discloses implementing its invention on a very common file system, *e.g.*, by modifying MicrosoftTM file systems via a MicrosoftTM-specified API, it is my opinion that a PHOSITA would be motivated to implement Aboulhosn in a similar way, which a PHOSITA would have expected to result in a working system.

3. Claim 2

180. Claim 2 depends from claim 1, which, in my opinion, is rendered obvious by Aboulhosn-Phillips. *See* Section XII.A.1. Claim 2 further recites **"wherein the individual software clients vary according to the individual**

operating systems of each of the multiple devices,” which, in my opinion, is disclosed by Aboulhosn.

181. Aboulhosn teaches that each member device includes a peer component that provides various file-sharing functionalities and “a conventional file system provided by an operating system.” Ex-1005, 5:14-15. In my opinion, a PHOSITA would have understood that each operating system requires its own file system that contains structures such as directories, metadata, file names, etc., and allows system calls such as open, close, read, write, etc. Thus, it is my opinion that, for Aboulhosn’s invention to be integrated with a conventional file system provided by one or more standard operating systems, Aboulhosn’s peer component necessarily needs to work with and vary according to the particular operating system.

182. This is confirmed by the ’448 Application (“incorporated by reference in its entirety” in Aboulhosn), which confirms that “[f]or each computer, the Local File Share Enabling Software is **necessarily adapted to the operating system** and file manager of the computer.” Ex-1009, pp.8-9, 6:30-7:2. Thus, in my opinion, a PHOSITA reading Aboulhosn and the incorporated ’448 Application would have appreciated that the peer component varies based on the operating system that it is running on.

183. To the extent it is found that Aboulhosn does not disclose this limitation, it is my opinion that Phillips discloses it. Phillips discloses that “[t]he manner by which the client node software is integrated with the operating system/native file system 48 may be **specific to each operating system/native file system 48** and is normally dictated by specification and application programming interfaces of the operating system/native file system creator.” *See* Ex-1006, 13:11-16. Therefore, it is my opinion that a PHOSITA would have also found it obvious to adapt Aboulhosn’s system to the specific operating system based on Phillips’s teachings and would have expected a reasonable expectation of success. *See* Ex-1006, 13:25-30 (“Those skilled in the art will appreciate how to modify the client node software for each operating system/native file system....”).

4. Claim 5

184. Claim 5 depends from independent claim 1, which, in my opinion, is rendered obvious by Aboulhosn-Phillips. *See* Section XII.A.1. Claim 5 further recites “**wherein the individual software clients facilitate creation of a visual representation of the virtual files on a remote one of each of the multiple devices that is indistinguishable from a visual representation of an actual file that is located on a local one of the multiple devices,**” which, in my opinion, is disclosed by Aboulhosn-Phillips for reasons I have explained in Section XII.A.1.i (1[H]).

5. Claim 6

185. Claim 6 depends from independent claim 1, which, in my opinion, is rendered obvious by Aboulhosn-Phillips. *See* Section IX.A. Claim 6 further recites “**wherein the meta-data includes at least information regarding file name, file type, file size, local path the file is stored in, date created, and date modified,**” which, in my opinion, is disclosed by Aboulhosn.

186. Aboulhosn discloses that “the metadata may identify the **file name**, the file owner, the **create date** of the file, the last **modified date** of the file, **the size of the file**, and so on.” Ex-1005, 2:28-30. In my opinion, a PHOSITA would understand that such metadata would also include the **local path that the file is stored in**. Most file systems available at the relevant time were (and still are) organized hierarchically via directories, wherein files can be placed within a series of directories or folders. Aboulhosn discloses such a hierarchical organization: “a shared folder may contain shared **subfolders**.” Ex-1005, 4:20-23. File systems consult file metadata to determine their local paths (*i.e.*, specific directories). *See* Ex-1029, p.3. Therefore, it is my opinion that a PHOSITA would have understood that any communication of metadata must include (or it would be obvious to include) the structure of the file system (*i.e.*, the local path of the files).

6. Claim 7

187. In my opinion, claim 7 is unpatentable because it is rendered obvious by Aboulhosn in view of Phillips. Claim 7 of the '338 Patent recites the following.

I address each of these limitations in my analysis below.

7[P]: A system for establishing a singular file system across multiple devices comprising:

7[A]: a web-accessible server;

7[B]: at least one software client plug-in;

7[C]: multiple devices for storing and operating on files;

7[D]: wherein the web-accessible server receives user information to open an account for establishing a singular file system across multiple devices, installs the at least one software plug-in on each of the multiple devices, accepts registration of the multiple devices, wherein the at least one software plug-in varies in accordance with an operating system of each of the multiple devices;

7[E]: further wherein the at least one software client plug-in scans its respective one of the multiple devices to inventory files thereon, creates a meta-index to files for the inventoried data, provides the meta-index of the inventoried data for its respective one of the multiple devices to the web-accessible server;

7[F]: further wherein the web-accessible server receives meta-indices from each of the multiple devices and integrates the meta-indices to create a single meta-index, provides the single meta-index and meta-indices for each of the other multiple

devices to each of the multiple devices via the individual software clients, integrates at each of the multiple devices metadata from the meta-indices of each of the other multiple devices into a local file system of each of the multiple devices to generate virtual files stored in the same locations as local files of the local file system, the virtual files indistinguishable from the local files by the local file system at each of the multiple devices, and continually updates the single meta-index on each of the multiple devices in response to operations on the files indexed thereon,

7[G]: wherein the individual software clients facilitate storage of the data within each of the multiple devices in accordance with the single meta-data index by modifying file systems of each of the multiple devices to include virtual files for data from the single meta-data index that is not local to a multiple device.

a. 7[P]

188. If the preamble is limiting, it is my opinion that Aboulhosn discloses it. As I explained in Section XII.A.1.a (preamble 1[P]), Aboulhosn discloses “[a] process for establishing a singular file system across multiple devices.” This process is performed by Aboulhosn’s system, which is illustrated in Figure 5 below.

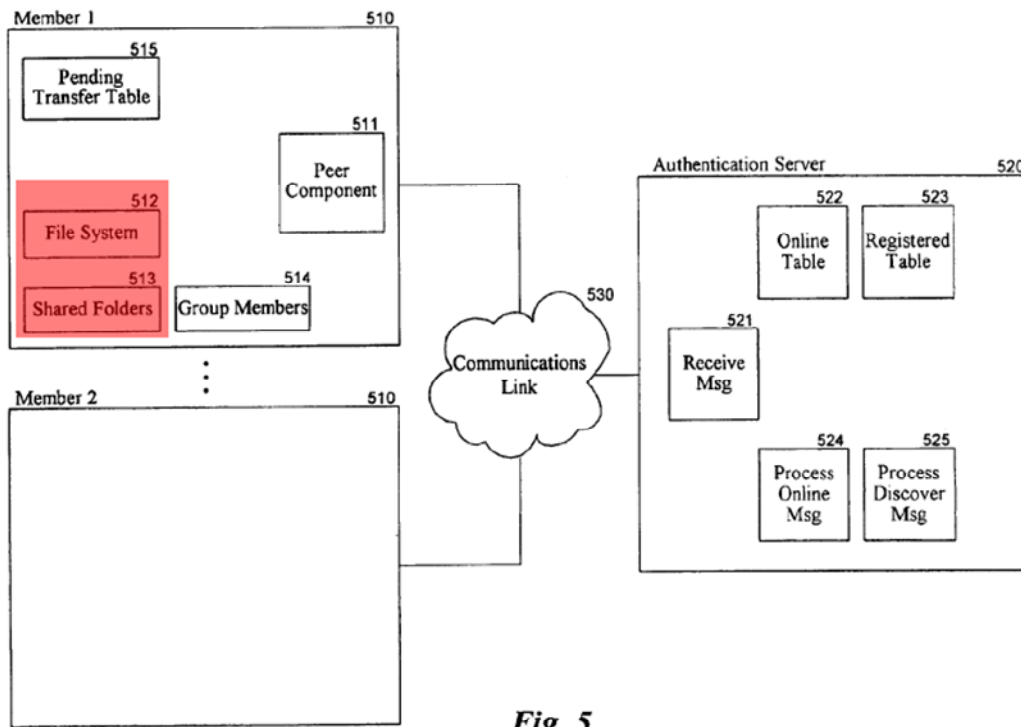


Fig. 5

Ex-1005, Fig. 5. I have annotated Aboulhosn’s singular file system in red.

b. 7[A]

189. Limitation 7[A] requires “a web-accessible server.” As explained in Section XII.A.1.b (1[A]), Aboulhosn’s GO computer is a server. Moreover, because Aboulhosn’s GO server communicates via the Internet, it is therefore a “web-accessible server.”

190. To the extent PO argues that Aboulhosn’s GO server is not “web-accessible,” it is my opinion that the GO server modified to use Phillips’s http/https protocol and/or host a web-server to communicate with a web browser,

discloses this limitation. Therefore, for the same reasons I explained above, Aboulhosn-Phillips renders obvious this limitation. *See* Section XII.A.1.b (1[A]).

c. 7[B]

191. Limitation 7[B] requires “at least one software client plug-in.”

192. In my opinion, both Aboulhosn and Phillips disclose a software client. *See* Section XII.A.1.c (1[B]). According to the '338 patent, a “plug-in” is merely a “form” “[t]he software client is provided in.” Ex-1001, 7:39-40. In general, a “plug-in” refers to “[a] small software program that **plugs into a larger application to provide added functionality.**” Ex-1012, p.3; *see also* Ex-1013, pp.3-4 (defining “plug-in” as “[a] program which is used in associated with a larger program, to provide the latter with additional functionality...Also called plug-in software, plug-in program, plug-in application, or plug-in component”).

193. In my opinion, Aboulhosn’s peer component is just such a “software client plug-in,” because it plugs into the member device’s file systems (i.e., by modifying file systems, *see* Section XII.A.1.k) and provides additional functionalities, such as “registering a computer system, placing a computer system online, creating groups, inviting computer systems to become members of a group, sending updates to metadata, and so on.” Ex-1005, 5:9-14.

194. To the extent PO argues that Aboulhosn does not disclose modifying file system, it is my opinion that Phillips discloses it. *See* Section XII.A.1.k. In

particular, as is the case with the '338 patent's plug-in, Phillips's "client node software **intervenes** and assists the operating system in identifying the appropriate file and in providing the data of the file to the operating system to complete the access to the [virtual] file." Ex-1006, 14:54-58; *c.f.* Ex-1001, 8:23-25 ("The software client described herein acts as an operating system plug-in to **intercept** calls made to files and programs in the file system of the respective user devices."). Therefore, in my opinion, Aboulhosn's peer component as modified by Phillips is a "software client plug-in" because it plugs into each member device's existing file system to provide additional functionalities. *See* Ex-1006, 11:22-12:20.

d. 7[C]

195. Limitation 7[C] requires "multiple devices for storing and operating on files." In my opinion, Aboulhosn discloses this limitation because, as I explained in Sections XII.A.1.a and XII.A.1.b (1[P] and 1[A]), Aboulhosn discloses multiple devices, e.g., member devices, such as M1, M2, and M3, which store and operate on files. *See, e.g.*, Ex-1005, Figs. 6B-6C.

e. 7[D]

196. As I show in table below, limitation 7[D] is substantially the same as the combination of limitations 1[A-C] and claim 2. Therefore, my opinion, limitation 7[D] is disclosed for the same reasons as limitations 1[A-C] and claim 2. *See* Sections XII.A.1.b-XII.A.1.d and XII.A.3.

Limitation 7[D]	Claim 1 and Claim 2	Sections
7[D][1]: the web-accessible server receives user information to open an account for establishing a singular file system across multiple devices,	1[A]: receiving user information to open an account for establishing a singular file system across multiple devices via a web-based system that includes at least one server;	XII.A.1.b
7[D][2]: installs the at least one software plug-in on each of the multiple devices,	1[B]: installing an individual software client on each of the multiple devices via the web-based system;	XII.A.1.c
7[D][3]: accepts registration of the multiple devices,	1[C]: accepting registration of multiple devices via the web-based system;	XII.A.1.d
7[D][4]: wherein the at least one software plug-in varies in accordance with an operating system of each of the multiple devices	Claim 2: wherein the individual software clients vary according to the individual operating systems of each of the multiple devices	XII.A.3

f. 7[E]

197. As I show in table below, limitation 7[E] is substantially the same as the combination of limitations 1[D-E]. Therefore, in my opinion, limitation 7[E] is disclosed for the same reasons as limitations 1[D-E]. See Sections XII.A.1.e-XII.A.1.f.

Limitation 7[E]	Claim 1	Sections
-----------------	---------	----------

7[E][1]: the at least one software client plug-in scans its respective one of the multiple devices to inventory files thereon, creates a meta-index to files for the inventoried data,	1[D]: scanning each of the multiple devices by each of the individual software clients to inventory data on each of the multiple devices and create a meta-index of the files for the inventoried data;	XII.A.1.e
7[E][2]: provides the meta-index of the inventoried data for its respective one of the multiple devices to the web-accessible server	1[E]: providing by the individual software clients via the multiple devices individual meta-indices of the inventoried data for each of the multiple devices to the at least one server	XII.A.1.f

g. 7[F]

198. As I show in table below, limitation 7[F] is substantially the same as the combination of limitations 1[E-I]. Therefore, in my opinion, limitation 7[F] is disclosed for the same reasons as limitations 1[E-I]. See Sections XII.A.1.f-XII.A.1.j.

Limitation 7[F]	Claim 1	Sections
7[F][1]: the web-accessible server receives meta-indices from each of the multiple devices and	1[E]: providing by the individual software clients via the multiple devices individual meta-indices of the inventoried data for each of the multiple devices to the at least one server ;	XII.A.1.f
7[F][2]: integrates the meta-indices to create a single meta-index ,	1[F]: integrating by the at least one server the individual meta-indices to create a single master meta-index ;	XII.A.1.g

7[F][3]: provides the single meta-index and meta-indices for each of the other multiple devices to each of the multiple devices via the individual software clients,	1[G]: providing by the at least one server the single master meta-index and meta-indices for each of the other multiple devices to each of the multiple devices via the individual software clients;	XII.A.1.h
7[F][4]: integrates at each of the multiple devices metadata from the meta-indices of each of the other multiple devices into a local file system of each of the multiple devices to generate virtual files stored in the same locations as local files of the local file system, the virtual files indistinguishable from the local files by the local file system at each of the multiple devices, and	1[H]: integrating metadata from the meta-indices of each of the other multiple devices into a local file system of each of the multiple devices to generate virtual files stored in the same locations as local files of the local file system, the virtual files indistinguishable from the local files by the local file system at each of the multiple devices; and	XII.A.1.i
7[F][5]: continually updates the single meta-index on each of the multiple devices in response to operations on the files indexed thereon	1[I]: continually updating the single master meta-index on the at least one server and each of the multiple devices in response to changes to the data indexed thereon	XII.A.1.j

h. 7[G]

199. Limitation 7[G] requires “**wherein the individual software clients facilitate storage of the data within each of the multiple devices in accordance with the single meta-data index by modifying file systems of each of the multiple devices to include virtual files for data from the single meta-data index that is not local to a multiple device,**” which is identical to limitation 1[J].

Therefore, in my opinion, limitation 7[G] is disclosed for the same reasons as limitation 1[J]. *See* Section XII.A.1.k.

7. Claim 8

200. Claim 8 depends from independent claim 7. As I explained above, all the limitations of claim 7 are rendered obvious by Aboulhosn-Phillips. Claim 8 further recites “**wherein the at least one software client plug-in facilitates storage of the files within its respective one of the multiple devices in accordance with the meta-data by modifying a file system of its respective one of the multiple devices to include virtual files for files from the single meta-index that are not local to the multiple device,**” which is substantially the same as limitation 7[G] and, in my opinion, is therefore disclosed for the same reasons. *See* Section XII.A.6.h.

8. Claim 9

201. Claim 9 depends from independent claim 8. As I explained above, all the limitations of claim 8 are rendered obvious by Aboulhosn-Phillips. Claim 9 further recites “**wherein the at least one software client plug-in facilitates creation of a visual representation of the virtual files on its respective one of the multiple devices that is indistinguishable from a visual representation of an actual file that is physically located on a another of the multiple devices,**”

which is substantially the same as claim 5 and, in my opinion, is therefore disclosed for the same reasons. *See* Section XII.A.4.

9. Claim 10

202. In my opinion, claim 10 is unpatentable because it is rendered obvious by Aboulhosn in view of Phillips. Claim 10 of the '338 Patent recites the following. I address these limitations in my analysis below.

10[P]: A non-transitory computer-readable storage medium storing a set of instructions that, when executed by a processor, cause the processor to perform operations, comprising:

10[A]: registering multiple devices with an account for establishing a singular file system across the multiple devices;

10[B]: scanning each of the multiple devices to inventory data on each of the multiple devices and create a meta-index of the files for the inventory data;

10[C]: generating a single master meta-index from meta-indices of files corresponding to inventory data on each of the multiple devices;

10[D]: providing the single master meta-index and meta-indices for each of the other multiple devices to each of the multiple devices, metadata from the meta-indices of each of the other multiple devices to be integrated into a local file system of each of the multiple devices to generate virtual files stored in the same locations as local files of the local file system, the virtual files indistinguishable from the local files by the local file system at each of the multiple devices; and

10[E]: facilitating storage of data within each of the multiple devices in accordance with the single master meta-index by modifying file systems of each of the multiple devices to include virtual files for data from the single master meta-data index that is not local to a multiple device.

a. 10[P]

203. To the extent that 10[P] is limiting, it is my opinion that Aboulhosn discloses it. Aboulhosn discloses that “[t]he computer systems and servers may include a **central processing unit, memory,...and storage devices (e.g., disk drives).**” Ex-1005, 5:50-53. Aboulhosn’s “memory... and storage devices (e.g., disk drives)” disclose a non-transitory computer-readable storage medium. Moreover, a central processing unit, known as CPU, is a processor. Ex-1021, p.1. Although Aboulhosn does not explicitly disclose it, a PHOSITA would have understood that the instructions that implement the file-sharing system is executed by Aboulhosn’s CPU.

204. Therefore, in my opinion, Aboulhosn discloses “[a] **non-transitory computer-readable storage medium** [e.g., computer-readable media, such as memory and storage devices] **storing a set of instructions that, when executed by a processor** [e.g., central processing unit], **cause the processor to perform operations** [e.g., instructions that implement the file sharing system].”

b. 10[A-E]

205. As I show in table below, limitations 10[A-E] collectively are substantially the same as limitations 1[A, C-D, F-J]. Therefore, in my opinion, limitation 10[A-E] are disclosed for the same reasons as limitations 1[A, C-D, F-J].

See Sections XII.A.1.b, XII.A.1.d-XII.A.1.e, and XII.A.1.g-XII.A.1.k.

Limitations 10[A-E]	Claim 1	Sections
10[A]: registering multiple devices with an account for establishing a singular file system across the multiple devices;	1[A]: receiving user information to open an account for establishing a singular file system across multiple devices via a web-based system that includes at least one server;	XII.A.1.b
	1[C]: accepting registration of multiple devices via the web-based system;	XII.A.1.d
10[B]: scanning each of the multiple devices to inventory data on each of the multiple devices and create a meta-index of the files for the inventory data;	1[D]: scanning each of the multiple devices by each of the individual software clients to inventory data on each of the multiple devices and create a meta-index of the files for the inventoried data;	XII.A.1.e
10[C]: generating a single master meta-index from meta-indices of files corresponding to inventory data on each of the multiple devices;	1[F]: integrating by the at least one server the individual meta-indices to create a single master meta-index ;	XII.A.1.g

10[D]: providing the single master meta-index and meta-indices for each of the other multiple devices to each of the multiple devices, metadata from the meta-indices of each of the other multiple devices to be integrated into a local file system of each of the multiple devices to generate virtual files stored in the same locations as local files of the local file system, the virtual files indistinguishable from the local files by the local file system at each of the multiple devices; and	1[G]: providing by the at least one server the single master meta-index and meta-indices for each of the other multiple devices to each of the multiple devices via the individual software clients	XII.A.1.h
	1[H]: integrating metadata from the meta-indices of each of the other multiple devices into a local file system of each of the multiple devices to generate virtual files stored in the same locations as local files of the local file system, the virtual files indistinguishable from the local files by the local file system at each of the multiple devices; and	XII.A.1.i
10[E]: facilitating storage of data within each of the multiple devices in accordance with the single master meta-index by modifying file systems of each of the multiple devices to include virtual files for data from the single master meta-data index that is not local to a multiple device	1[J]: wherein the individual software clients facilitate storage of the data within each of the multiple devices in accordance with the single meta-data index by modifying file systems of each of the multiple devices to include virtual files for data from the single meta-data index that is not local to a multiple device	XII.A.1.k

10. Claim 11

206. Claim 11 depends from independent claim 10, which, in my opinion, is rendered obvious by Aboulhosn-Phillips. *See* Section XII.A.9. Claim 11 further recites “**continually updating the single master meta-index on each of the**

multiple devices in response to changes to the data indexed thereon,” which is substantially the same as limitation 1[I] and, in my opinion, is therefore disclosed for the same reasons. *See* Section XII.A.1.j.

11. Claim 12

207. Claim 12 depends from independent claim 10, which, in my opinion, is rendered obvious by Aboulhosn-Phillips. *See* Section XII.A.9. Claim 12 further recites “**updating the single master meta-index on each of the multiple devices in response to changes to meta-indices of a local file of one of the multiple devices,**” which is substantially the same as limitation 1[I] except that claim 12 requires “updating the single master meta-index on each of the multiple devices in response to **changes to meta-indices** of a local file” as opposed to “changes to the data.”

208. In my opinion, a PHOSITA would have understood that modifying an existing file in the Aboulhosn-Phillips system necessarily changes its associated metadata. As detailed in Section XII.A.5 (claim 6), the meta-data includes “file name, file type, file size...date created, and date modified.” Ex-1001, claim 6. Therefore, when a local file’s content is modified, its “date modified,” and possibly “file size,” changes. When a local file’s file name is modified, its “file name” changes. Therefore, claim 12 is disclosed for the same reasons explained in

Section XII.A.1.j (1[I]). Therefore, it is my opinion that claim 12 is disclosed for the same reasons I explained in Section XII.A.1.j (limitation 1[I]).

12. Claim 13

209. Claim 13 depends from independent claim 10, which, in my opinion, is rendered obvious by Aboulhosn-Phillips. *See* Section XII.A.9. Claim 13 further recites **“updating the single master meta-index on each of the multiple devices in response to changes to a local file of a first device, the local file corresponding to a virtual file of a second device, the local file system of the second device retrieving the local file from the first device.”**

210. As I explained in Section XII.A.1.j (limitation 1[I]), Aboulhosn discloses “updating the single master meta-index on each of the multiple devices in response to changes to a local file of a first device” required by claim 13.

211. Aboulhosn also teaches “the local file corresponding to a virtual file of a second device.” *See, e.g.,* Ex-1005, 2:7-8 (“virtual files identifying the actual file that is stored at another member”); Figs. 2-4.

212. Finally, Aboulhosn discloses “the local file system of the second device retrieving the local file from the first device.” *See, e.g.,* Ex-1005, claim 22 (“when the virtual file is accessed, retrieving the file stored at the second computer system using the metadata associated with the virtual file.”).

213. Therefore, it is my opinion that Aboulhosn-Phillips discloses claim 13.

13. Claim 14

214. Claim 14 depends from independent claim 10, which, in my opinion, is rendered obvious by Aboulhosn-Phillips. *See* Section XII.A.9. Claim 14 further recites “**wherein the virtual files and the local files in the first device are indistinguishable by the local file system of the first device,**” which is substantially the same as limitation 10[D] and, in my opinion, is therefore disclosed for the same reasons.

B. My Opinions as to Claims 3 and 4 based on Aboulhosn-Phillips in view of Tsang

1. Claim 3

215. Claim 3 depends from claim 1. As I explained above, all the limitations of claim 1 are rendered obvious by Aboulhosn-Phillips. Claim 3 further recites “**wherein registration of the multiple devices includes selection of categories of data for each of the multiple devices to be inventoried,**” which, in my opinion, is disclosed by Aboulhosn-Phillips in view of Tsang.

216. As detailed in Section XII.A.1.d (1[C]), Aboulhosn discloses accepting registration of multiple devices. Aboulhosn also discloses selecting data, *e.g.*, files, to add to the group folder. *See* Section XII.A.1.e (1[D]); *see also* Ex-1009, p.12, 10:9-11 (“the user...select[s] with his local file manager one or more

files...to upload or push to other users in the group.”). While Aboulhosn does not disclose selecting “categories of data” when registering a device, Tsang discloses it in Figure 6: “a process 600 for creating partnership data that occurs **when a partner computing device is added to an account.**” Ex-1008, 6:59-61. As part of this process, Tsang discloses that “[a]t block 610 [yellow], a **list of local folders may be identified.** The list **identifies the folders to include** (*i.e.*, ‘share’) in the combined file space and their corresponding mapped locations.”

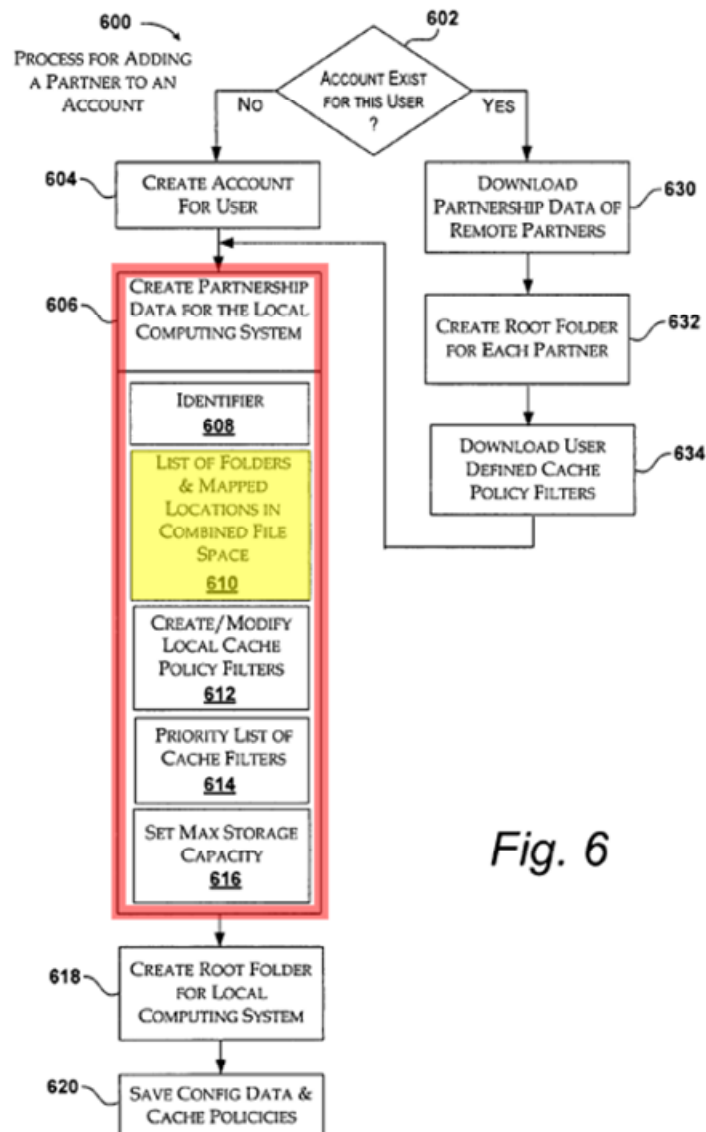


Fig. 6

Ex-1008, Fig. 6. In other words, a PHOSITA would have recognized that Tsang discloses that during registration of an individual device, selection of local folders is performed, such that the selected folders are shared in the combined file space, thereby adding the files within these selected folders to the combined file space. In the Aboulhoshn-Phillips system, once these folders (and files contained therein) are

selected to be added to the shared folder, these files are inventoried to generate a meta-index. *See* Section XII.A.1.e (limitation 1[D]).

217. Moreover, just like the '338 patent's categories of data (*i.e.*, “documents...pictures...audio...video” (Ex-1001, 5:49-53)), Tsang discloses that “the list of **folders** may [] **include...** ‘**Documents**’, ‘**Music**’, ‘**Pictures**’, and ‘**Videos**’” (Ex-1008, 7:18-21). Because these folders, as their names suggest, correspond to different **categories** of data, it is my opinion that a PHOSITA would have understood that by selecting “Documents” folder, the selection is of a category of documents, by selecting “Music” folder, the selection is of a category of music, by selecting “Pictures” folder, the selection is of a category of pictures, and by selecting “Videos” folder, the selection is of a category of videos.

218. I note that claim 3 does not specify who performs the “selection of categories of data.” The '338 patent specification states that the selection may be performed either **by the user manually** or **by the member device automatically**. Ex-1001, 5:46-56 (stating that “for each individual user device, **the user...selects** the type or category of data, e.g., documents (exemplary: text), pictures (exemplary: .jpg, .img), audio (exemplary: MP3), video (exemplary: .dvi, files .mov files) available on the device...” and that “[i]n a preferred embodiment, all categories are **automatically selected**. Alternatively, this selection may be

accomplished by a drop down menu listing file types, categories, extensions, etc.”). In my opinion, both possibilities are disclosed by Tsang.

219. **Tsang Discloses Manual Selection.** Specifically, Tsang discloses that “[a]t block 606 [red]...a user interface element may be displayed for inputting the partnership data.” Ex-1008, 7:7-10.

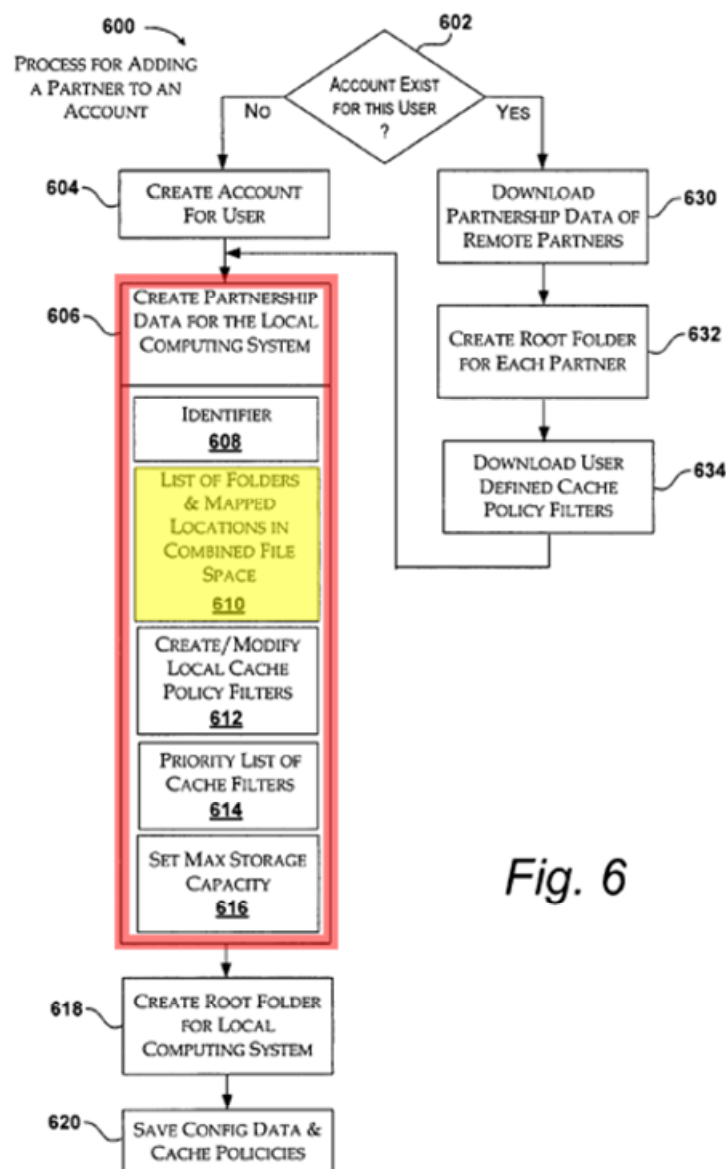


Fig. 6

Ex-1008, Fig. 6. Therefore, in my opinion, a PHOSITA would have understood that the user interface may also be used for the user to select a list of folders at block 610 (yellow), which is part of the partnership data creation process 606 (red). In other words, Tsang discloses selecting categories of data by the user manually.

220. **Tsang Discloses Automatic Selection.** On the other hand, like the '338 patent, Tsang discloses that “[b]y default, the list of folders may **automatically include known folders** that exist on the partner computing device. For example, “Documents”, “Music”, “Pictures”, and “Videos” may be **automatically mapped** into the combined file space.” Ex-1008, 7:16-20. Thus, in my opinion, Tsang also discloses selecting categories of data by an individual device automatically.

221. Therefore, in my opinion, Aboulhosen-Phillips in view of Tsang discloses that “**registration of the multiple devices includes selection of categories of data** [e.g., Tsang’s selection of folders, such as “Documents”, “Music”, “Pictures”, and “Videos”] **for each of the multiple devices to be inventoried**” required by claim 3.

2. Claim 4

222. Claim 4 depends from independent claim 3, which, in my opinion, is rendered obvious by Aboulhosen-Phillips-Tsang. *See* Section XII.B.1. Claim 4 further recites “**wherein categories of data include documents, videos, pictures**

and music,” which, in my opinion, is disclosed by Tsang for the same reasons I explained in Section XII.B.1 (claim 3).

3. Motivation to Combine Aboulhosn-Phillips with Tsang

223. In my opinion, Tsang, Aboulhosn, and Phillips qualify as **analogous art** at least because they are in the **same field of endeavor** as the '338 patent, *i.e.*, file sharing and synchronization. Like Aboulhosn, Phillips, and the '338 patent, Tsang is in CPC class G06F and U.S. class 707. Also like Aboulhosn, Phillips, and the '338 patent, Tsang discloses a “a combined file space for files from multiple partner computing devices” (Ex-1008, Abstract), and a user is able to view “ghosted files” that “do not have content of the file stored in the local combined file space, but rather have metadata associated with the file” (*id.*, 3:50-53).

224. Like Aboulhosn, Phillips, and the '338 patent, Tsang discloses registering devices to a combined file space. Ex-1008, Abstract; Ex-1001, 5:43-46; Ex-1005, Fig.8; Ex-1006, Figs. 5, 6A-6B. While Aboulhosn does not disclose selecting categories of data when registering a device, it is my opinion that a PHOSITA would have been motivated to incorporate Tsang’s folder (category) selection process as part of device registration such that a user registering a device is given an opportunity to select the data he/she wants to share at the initial setup

stage. As described in all references and in the '338 patent, a user registers a new device to the shared group not only to view files from other devices but also to share files with other devices. Thus, a PHOSITA would have recognized that by providing the user an opportunity to select the data he/she wants to share when registering the device, the user does not need to wait until device registration is complete to upload such data. Instead, the data upload may be performed automatically upon completing device registration, thereby consuming less computer resources, saving unnecessary user operations, and improving user experience. Moreover, like Aboulhosn, Tsang discloses that “[i]n order to provide the same view of the shared files in a combined file space with limited space, the combined file space may include a ‘ghosted file’...instead of actual content...for the associated file.” Ex-1008, 3:43-47. As a first step, Tsang teaches selecting categories of data (610, yellow) when initially setting up a device.

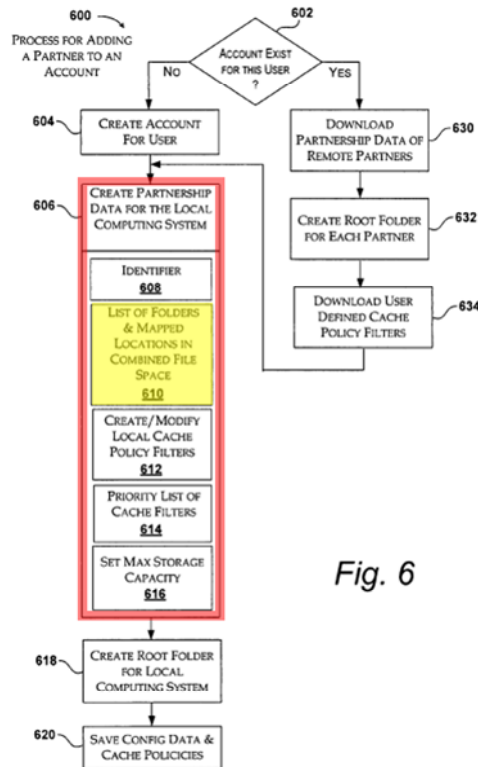


Fig. 6

Ex-1008, Fig. 6. In particular, a PHOSITA would have recognized that Tsang discloses applying and prioritizing cache policy filters to these categories of data to determine which state (*e.g.*, ghosted, cached, master), as illustrated in Figure 5 below (music in yellow, documents in green, videos in blue). *Id.*, 5:66-67, 6:30-37, Fig. 3.

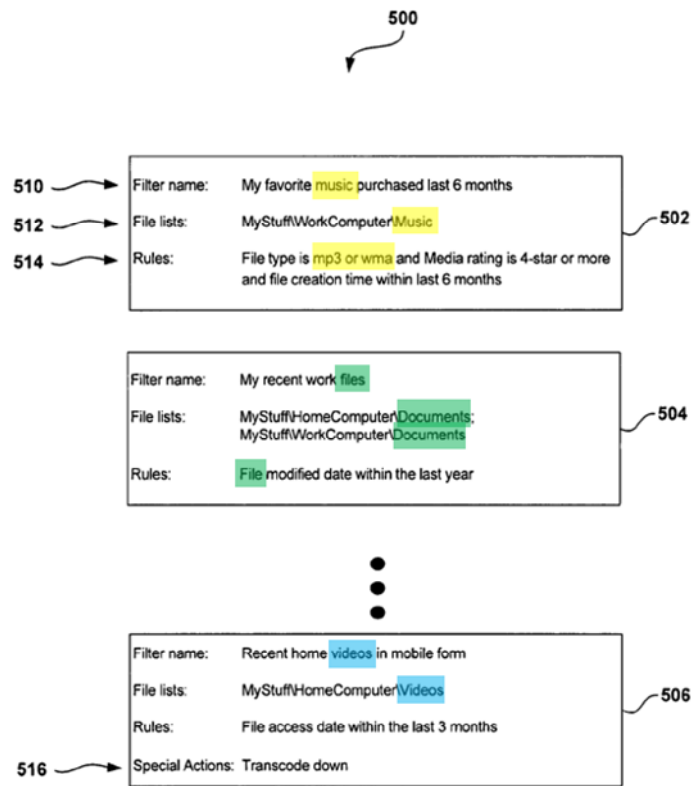


Fig. 5

225. Therefore, it is my opinion that a PHOSITA implementing Aboulhosn's system would have been motivated to incorporate Tsang's data category selection feature to facilitate virtual/local file state determination and control, thereby enabling user access to the entire shared folder even on devices with limited space, as contemplated in both Aboulhosn and Tsang. Finally, by being able to select multiple folders (and thereby categories of data) to share, a PHOSITA would have recognized that the user does not need to navigate among multiple folders to select individual files to add to Aboulhosn-Phillips's shared

folder, which further reduces unnecessary and cumbersome user operations and improves user experience.

226. In my opinion, A PHOSITA would have had a **reasonable expectation of success** in incorporating Tsang's folder/data category selection as part of Aboulhosn-Phillips's device registration process. Tsang discloses two ways of selecting categories: through (1) a user interface, and (2) automatic selection of default folders or categories of data. Ex-1008, 7:9-11 ("In one embodiment, a user interface element may be displayed for inputting the partnership data."); 7:18-20 ("By default, the list of folders may automatically include known folders that exist on the partner computing device."). It was well known at the time of the '338 patent to utilize a user interface to select folders to share with other devices. *See, e.g.,* Ex-1023, Figs. 7-9. In my opinion, a PHOSITA would also have understood that selecting default folders/categories of data also involves simple and well-known modifications to Aboulhosn-Phillips's device registration process on a member device and would have resulted in a working system.

C. **My Opinions as to Claims 1-2 and 5-14 in view of the Aboulhosn-Phillips-Shappell combination**

1. **Claims 1-2, 5, and 7-14**

227. To the extent PO argues that Aboulhosn-Phillips does not explicitly disclose “modifying file systems” required by 1[J] and 7[G], it is my opinion that Shappell discloses it.

228. Like Aboulhosn, Shappell discloses generating and displaying virtual files on a remote device. Ex-1007, ¶79 (“a display icon 1504 to denote a ‘missing’ file that is available on another system...is ghosted or semi-transparent.”).

229. Moreover, Shappell discloses that its “invention may be **implemented as a file system driver or filter driver that resides below the file system and accesses file system services** provided by an operating system.” Ex-1007, ¶44. In my opinion, a PHOSITA would have understood that implementing “a file system driver or filter driver that resides below the file system and accesses file system services” means embedding a file system driver or filter driver between the file system and file system services to modify the way the file system would otherwise behave, *i.e.*, modifying the file system. This is consistent with a PHOSITA’s understanding of a file system driver or filter driver. Ex-1030, p.5 (“a filter driver is a **kernel-mode driver**. It is developed primarily to **intercept requests** targeted to an existing kernel-mode driver, to **allow the addition of new functionality**

beyond what is currently available.”), p.6 (“Therefore, if you want to provide some software that will extend, **modify**, or completely supplant an existing module and if you wish to maintain **complete transparency** to the user when providing your specialized functionality, consider designing a filter driver.”).

230. In fact, computers are built from abstractions (Ex-1028), wherein controlling complicated components (such as computer hardware) is made simple by exposing only what a component does (and hiding how it does it). These abstractions are naturally organized into “layers” with the most complicated components (i.e., hardware) at the bottom and the simplest components at the top (i.e., programs). *Id.* When a component is described as existing “below” another component, that is to say that the functionality of the component above relies on the component below. Thus, a file system driver that resides “below” the file system, *e.g.*, between the file system and its file system services, modifying the way the file system would otherwise behave.

231. Moreover, to the extent PO argues that Aboulhosn-Phillips does not explicitly disclose claim 7’s “software client plug-in,” it is my opinion that Shappell discloses it. In my opinion, a PHOSITA would also have understood that Shappell’s file system filter driver is a software client plug-in because it plugs in between the member device’s existing file system and its file system services, modifying the way the file system would otherwise behave, *e.g.*, by intercepting

requests targeted at a file system or another file system filter driver and extend or replace functionality provided by the original target of the request.

232. I note again that the '338 patent describes a software plug-in as follows: “The software client described herein act as **an operating system plug-in to intercept calls** made to files and programs in the file system of the respective user devices.” Ex-1001, 8:23-25; *see also id.*, 7:39-42 (“The software client is provided in the form of an operating system extension, or ‘plug-in’ that is unique to the type of device/operating system which is established at the time of registration.”). Notably, “operating system extension,” “[operating system] plug-in”, “device driver”, “kernel module” all refer to the same piece of technology: they modify the operating system in some way (often to extend features for unique devices). Shappell’s file system driver also modifies the operating system (via intercepting, etc.), and thus, is a “plug-in” with the very definition used in the '338 patent.

2. Motivation to Combine Aboulhosn-Phillips with Shappell

233. In my opinion, Shappell, Aboulhosn, and Phillips qualify as **analogous art** at least because they are in the **same field of endeavor** as the '338 patent, *i.e.*, file sharing and synchronization. Like Aboulhosn, Phillips, and the '338 patent, Shappell is in CPC class G06F. Also like Aboulhosn, Phillips, and the

'338 patent, Shappell discloses a “shared space” for multiple users (Ex-1007, Abstract), and a user is able to view files that are “on another system” and “not present locally.” *Id.*, ¶79.

234. In my opinion, a PHOSITA would have been motivated to combine Shappell with Aboulhosn-Phillips. As mentioned above, both the Aboulhosn-Phillips system and Shappell discloses file sharing and synchronization among multiple users and for files that are not local to an individual device, the files are displayed as “virtual files” (in Aboulhosn-Phillips) or “ghosted” (in Shappell). As explained above, both Aboulhosn and Phillips contain sufficient disclosure to indicate to a PHOSITA in which computer system layer to implement the Aboulhosn-Phillips system, e.g., modifying the conventional file system. *See* Section XII.A.1.k (1[J]). However, to the extent they do not, it is my opinion that a PHOSITA would have been motivated to look to references with similar systems, such as Shappell, for guidance as to the appropriate computer system layer for implementing the Aboulhosn-Phillips system. Moreover, Shappell provides a system where files are “managed quickly and effectively.” Ex-1007, ¶11. Therefore, in my opinion, a PHOSITA looking to make Aboulhosn’s system quicker and more effective would have been motivated to implement it as a file system driver or filter driver as taught by Shappell.

235. In my opinion, a PHOSITA would have had a **reasonable expectation of success** in implementing the Aboulhosn-Phillips system as a file system driver or filter driver that resides below the file system as taught by Shappell. As discussed above, the Aboulhosn-Phillips system and Shappell's system are analogous and both have the capability to display and allow access to virtual/ghosted files that are stored remotely. Ex-1005, Abstract, Figs. 2-4; Ex-1007, Fig.15, ¶82. Therefore, in my opinion, a PHOSITA would have understood that Shappell's particular computer system layer, e.g., "a file system driver or filter driver that resides below the file system," that is used to implement Shappell's system would have also been suitable for implementing Aboulhosn-Phillips's system and have resulted in a working system.

3. Claim 6

236. To the extent PO argues that Aboulhosn does not disclose metadata including "local path the file is stored in" required by claim 6, it is my opinion that Shappell discloses it. Aboulhosn indicates that the metadata may include more information than enumerated—noting the "and so on." *Id.* On the other hand, Shappell discloses various metadata that includes "Path to file," in addition to "Name of file," "size of the file," and "modification time." Ex-1007, ¶78. Thus, in my opinion, it would have been obvious for a PHOSITA to include additional information, such as Shappell's "Path to file," in Aboulhosn's metadata.

Moreover, Aboulhosn discloses that “when a member accesses a virtual file,... the file owner retrieves the data of the file and sends the data to the accessing member.” Ex-1005, 7:16-24. Therefore, it is my opinion that a PHOSITA looking to make Aboulhosn’s system quicker and more effective as Shappell’s system (Ex-1007, ¶11) would have been motivated to incorporate Shappell’s file path in the metadata to enable faster and more accurate data retrieval. In fact, it was common practice to include local file paths in the metadata. *See* Section IX.E. Adding additional information to metadata was simple and well-known operations in file system implementations and would have resulted in a working system.

D. My Opinions as to Claims 3 and 4 in view of the Aboulhosn-Phillips-Tsang in view of Shappell

237. As I explained in Section XII.B (Ground 2), Aboulhosn-Phillips-Tsang discloses claims 3 and 4. In my opinion, a PHOSITA looking to implement the Aboulhosn-Phillips-Tsang system and wishing to do so in a manner that enables quick and effective file management and retrieval would have been motivated to look to a reference, such as Shappell, which discloses that its system enables files to be “managed quickly and effectively.” Ex-1007, ¶11. A PHOSITA would have therefore been motivated to implement the Aboulhosn-Phillips-Tsang system in the same or similar way as disclosed by Shappell—*i.e.*, as a file system driver or filter driver, as I explained in Section XII.C.2.

238. Therefore, in my opinion, Aboulhosn-Phillips-Tsang in view of Shappell renders claims 3 and 4 obvious.

XIII. CONCLUDING STATEMENTS

239. In my opinion, the challenged claims 1-2, 5-6, 7-9, 10-14 are obvious by Aboulhosn in view of Phillips. As such, in my opinion, these claims should be found unpatentable and cancelled.

240. In my opinion, claims 3 and 4 are obvious by Aboulhosn in view of Phillips and further in view of Tsang. As such, in my opinion, these claims should be found unpatentable and cancelled.

241. In my opinion, the challenged claims 1-2, 5-6, 7-9, 10-14 are obvious by Aboulhosn in view of Phillips and Shappell. As such, in my opinion, these claims should be found unpatentable and cancelled.

242. In my opinion, claims 3 and 4 are obvious by Aboulhosn in view of Phillips and Tsang and further in view of Shappell. As such, in my opinion, these claims should be found unpatentable and cancelled.

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

and that such willful false statements may jeopardize the results of these proceedings.

Date: December 12, 2023

A handwritten signature in black ink, appearing to read "Patrick D. McDaniel", written in a cursive style.

Patrick D. McDaniel, Ph.D.