UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

UNIFIED PATENTS, INC.,
SAP AMERICA INC.,
Petitioners,

v.

CLOUDING IP, LLC,
Patent Owner.

Case IPR2013-00586
Case IPR2014-00306
Patent 6,738,799 B2

Before JAMESON LEE, JUSTIN BUSCH, and RAMA G. ELLURU,
Administrative Patent Judges.

BUSCH, *Administrative Patent Judge.*

FINAL WRITTEN DECISION
35 U.S.C. § 318(a) and 37 C.F.R. § 42.73

I. BACKGROUND

Unified Patents, Inc. ("Unified") filed a Petition (Paper 1, "Pet.") requesting *inter partes* review of claims 1, 5–10, 12, 16–21, 23, 24, 30, 31, 37, and 42 (the "challenged claims") of U.S. Patent No. 6,738,799 B2 ("the '799 Patent") under 35 U.S.C. §§ 311–319. On March 21, 2014, the Board instituted an *inter partes* review of the challenged claims on six asserted grounds of unpatentability ("Dec. on Inst."). Paper 9. On December 27, 2013, SAP America, Inc. ("SAP") filed a petition (the "SAP Petition"), asserting the same grounds (against the same claims) as asserted by Unified in the Petition. On May 20, 2014, the Board instituted an *inter partes* review of the challenged claims and joined the review based on the SAP Petition with this *inter partes* review. Paper 17. Subsequent to institution and joinder of the two reviews, Clouding IP, LLC ("Patent Owner") filed a Patent Owner Response ("PO Resp.") responding to the petitions filed by Unified and SAP (collectively, "Petitioners"). Paper 18. Patent Owner also filed a Contingent Motion to Amend ("MTA" or "Mot. to Amend"). Paper 19. Petitioners filed a Reply (Paper 22, "Pet. Reply") to the Patent Owner Response and an Opposition (Paper 23, "Opp. MTA") to the Contingent Motion to Amend. Patent Owner filed a Reply to Petitioners' Opposition to the MTA ("PO Reply"). Paper 25. Oral hearing was held on October 16, 2014.[1]

The Board has jurisdiction under 35 U.S.C. § 6(c). This final written decision is issued pursuant to 35 U.S.C. § 318(a) and 37 C.F.R. § 42.73. For the reasons that follow, we determine that Petitioners have shown by a preponderance of the evidence that the challenged claims are unpatentable. Patent Owner's Contingent Motion to Amend is denied.

[1] The record includes a transcript of the oral hearing ("Hr'g Tr."). Paper 35.

A. The ’799 Patent (Ex. 1001)

The ’799 Patent is related to a method for file synchronization using a signature list. Ex. 1001, Title. In particular, the ’799 Patent discloses a method for synchronizing the local copies of files on client computers to the current versions of the files on a network drive. Ex. 1001, 1:24–27. According to the ’799 Patent, an object of the method is to provide a mechanism by which a user can be provided automatically with a current version of a subscription file in an efficient manner. Ex. 1001, 3:36–41. This is accomplished by having a server computer monitor network files for changes, and then send users email notifications and updates when there is a change to the files. Ex. 1001, 3:41–44.

B. Illustrative Claim

Of the challenged claims, claims 1, 12, 23, 30, 37, and 42 are independent claims. Claim 1 is similar to claim 23, with the exception that claim 1 includes an additional limitation (“wherein the new segment . . .”) not present in claim 23. Claims 1, 23, and 37 are method claims. Claims 12, 30, and 42 are computer readable media versions of claims 1, 23, and 37, respectively. Thus, claims 1 and 37 are exemplary of the claimed subject matter, and are reproduced below (emphases added):

1. A method for a first computer to generate an update for transmission to a second computer that permits the second computer to generate a copy of a current version of a file comprised of a first plurality of file segments from a copy of an earlier version of the file comprised of a second plurality of file segments, such that each file segment corresponds to a portion of its respective file, the method comprising the steps of:

for each segment of the current version of the file,

(a) searching an earlier version of a signature list corresponding to an earlier version of the file for an old segment signature which matches a new segment signature corresponding to the segment;

(b) if step (a) results in a match, writing ***a command*** in the update for the second computer ***to copy*** an old segment of the second computer's copy of the earlier version of the file into the second computer's copy of the current version of the file, wherein the old segment corresponds to the segment for which a match was detected in step (a); and

(c) if step (a) results in no match, writing ***a command*** in the update for the second computer ***to insert*** a new segment of the current version of the file into the second computer's copy of the current version of the file;

wherein the new segment of the current version of the file is written into the update and the unchanged segment is excluded from the update; and

wherein steps (a) through (c) are performed by the first computer, without interaction with the second computer, in response to the first computer detecting a change between the current version of the file and the earlier version of the file.

> 37. A method for a first computer to provide updates for transmission to a second computer that permits the second computer to obtain most recent versions of files, the method comprising the steps of:
>
> (a) determining whether the second computer has ***a latest version*** of a file, wherein said determining is performed by the first computer without interaction with the second computer;
>
> (b) generating an update, if the second computer does not have a latest version of the file, wherein said generating is performed by the first computer without interaction with the second computer; and
>
> (c) transmitting the update from the first computer to the second computer.

C. Related Proceedings

Petitioners indicate that the ’799 Patent was the subject of the following terminated *inter partes* reviews before the Board: Oracle Corp. v. Clouding IP, LLC, IPR2013-00073[2] and Oracle Corp. v. Clouding IP, LLC, IPR2013-00261. Pet. 4. Petitioners indicate that the ’799 Patent is the subject of the following co-pending federal district court cases: *Clouding IP, LLC v. EMC Corp., et al.*, Case No. 1:13-cv-01455 (D. Del.); *Clouding IP, LLC v. Dropbox Inc.*, Case No. 1:13-cv-01454 (D. Del.); *Clouding IP, LLC v. SAP AG, et al.*, Case No. 1:13-cv-01456 (D. Del.); *Clouding IP, LLC v. Verizon Inc.*, Case No. 1:13-cv-01458 (D. Del.); *Clouding IP, LLC v. Rackspace, Hosting Inc.*, Case No. 1:12-cv-00675 (D. Del.); *Clouding IP, LLC v. Amazon.com Inc.*, Case No. 1:12-cv-00641 (D. Del.); *Clouding IP, LLC v. Oracle Corp.*, Case No. 1:12-cv-00642 (D. Del.); *Clouding IP, LLC*

[2] Petitioners identify IPR2012-00073 as a related matter. Pet. 4. However, IPR2013-00073 is the related *inter partes* review involving the ’799 Patent.

v. Google Inc., Case No. 1:12-cv-00639 (D. Del.). Pet. 4. Petitioners indicate that the '799 Patent also was the subject of the following terminated federal district court cases: *Clouding IP, LLC v. Apple Inc.*, Case No. 1:12-cv-00638 (D. Del.); and *Clouding IP, LLC v. Microsoft Corp.*, Case No. 1:12-cv-00640 (D. Del.). Pet. 4.

D. Asserted Grounds of Unpatentability

The Board instituted *inter partes* review on the following asserted grounds of unpatentability under 35 U.S.C. §§ 102 and 103 (Dec. on Inst. 29–30):

References	Basis	Challenged Claim(s)
Williams[3]	§ 102(e)	1, 12, 23, 24, 30, 31, 37, and 42
Williams and Miller[4]	§ 103(a)	5–10 and 16–21
Balcha[5]	§ 102(e)	37 and 42
Balcha and Miller	§ 103(a)	1, 5, 9, 10, 12, 16, 20, 21, 23, 24, 30, and 31
Balcha, Miller, and Freivald[6]	§ 103(a)	6–8 and 17–19
Balcha and Freivald	§ 103(a)	37 and 42

II. ANALYSIS

A. Claim Construction

The Board interprets claims of an unexpired patent using the broadest reasonable construction in light of the specification of the patent. 37 C.F.R.

[3] U.S. Patent No. 5,990,810, issued Nov. 23, 1999 (Ex. 1006) ("Williams").
[4] U.S. Patent No. 5,832,520, issued Nov. 3, 1998 (Ex. 1004) ("Miller").
[5] U.S. Patent No. 6,233,589 B1, issued May 15, 2001 (Ex. 1003) ("Balcha").
[6] U.S. Patent No. 5,898,836, issued Apr. 27, 1999 (Ex. 1005) ("Freivald").

§ 42.100(b); *see* Office Patent Trial Practice Guide, 77 Fed. Reg. 48,756, 48,766 (Aug. 14, 2012). Claims are to be given their broadest reasonable interpretation consistent with the specification, reading the claim in light of the specification as it would be interpreted by one of ordinary skill in the art. *In re Am. Acad. of Sci. Tech. Ctr.*, 367 F.3d 1359, 1364 (Fed. Cir. 2004).

Although the parties argue other terms, for purposes of this decision, we find it necessary to construe explicitly only "update," "writing a command . . . to copy," and "determining whether the second computer has a latest version of a file, wherein said determining is performed by the first computer without interaction with the second computer." We provide the bases for the construction of each of the explicitly construed terms below.

1. "update"

Petitioners propose construing "update" as "information for updating a file or an up-to-date version of a file." Pet. 15 (citing Ex. 1010, 10). Patent Owner does not propose a construction for "update" in its Patent Owner Response.

The claim term "update" has the following dictionary definition: "current information for updating something" or "*an up-to-date version*, account, or report."[7] We do not find an explicit definition in the Specification of the '799 Patent for an update, nor do we find anything in the Specification inconsistent with a construction encompassing an up-to-date version of a file. Therefore, in the context of file synchronization, we construe the claim term "update" broadly, but reasonably, as information for updating a file or an up-to-date version of a file.

[7] MERRIAM-WEBSTER DICTIONARY, http://www.merriam-webster.com/dictionary/update (last visited Feb. 11, 2014) (emphasis added).

2. *"writing a command . . . to copy"*

Each of claims 1, 12, 23, and 30 recites the following claim phrase: "writing *a command* in the update for the second computer *to copy* an old segment of the second computer's copy of the earlier version of the file into the second computer's copy of the current version of the file." Ex. 1001, claims 1, 12, 23, 30 (emphasis added). Hereinafter, we refer to this claim phrase as "writing a command . . . to copy."

Petitioners propose construing "a command to copy" as "an instruction that causes the computer to duplicate information or data." Pet. 15 (citing Ex. 1010, 11). Patent Owner asserts a "plain meaning reading of [the writing a command . . . to copy phrase] demands that a *command to copy* be written in the update that is used by the second computer to generate a copy of the current version of the file." PO Resp. 10.

We note that the recited "writing a command . . . to copy" language merely requires that a command, which causes the second computer to copy a portion of an earlier version of a file into a current version of the file, be written in the update. The claim does not limit the command to a specific format. Therefore, we broadly, but reasonably, construe "writing a command . . . to copy" as inserting an instruction into the update that causes the second computer to duplicate information or data from an earlier version of a file into a current version of a file.

3. *"determining whether the second computer has a latest version of a file, wherein said determining is performed by the first computer without interaction with the second computer"*

Neither Petitioners nor Patent Owner have proposed explicitly a construction of "determining whether the second computer has a latest version of a file, wherein said determining is performed by the first computer without interaction with the second computer" (the "determining limitation"). We find the language of the determining limitation is sufficiently clear that the plain and ordinary meaning is the proper construction. Nevertheless, because the determining limitation is central to the disputes in this proceeding, we address two key components of the determining limitation to explain how an ordinarily skilled artisan would have understood the determining limitation when read in the context of the Specification of the '799 Patent at the time of invention. In particular, we will discuss how an ordinarily skilled artisan would have understood what the latest version of a file is, and what it means for the determination to be "performed by the first computer *without interaction with the second computer*."

Patent Owner appears to argue that the determining limitation requires the first computer to be absolutely certain that the version on the second computer is not the most recent version of the file. *See, e.g.*, PO Resp. 7–8 (arguing that claims 37 and 42 do not allow for the possibility that a more recent file would be overwritten by an older file), 38 (distinguishing prior art where "computer E1 cannot be certain whether or not computer E2 stores a latest version of the subject file").

In its arguments against Petitioners' challenges, Patent Owner provides examples of how uncertainty could occur in the prior art. *See, e.g.*, PO Resp. 7 ("Balcha only discloses detecting whether a base file has been modified, regardless of when that modification may have occurred relative to other copies of the same base file"), 25 ("Freivald is unconcerned with whether or not a subscribed client has a latest version of a Web page (or any version at all for that matter), and sends notifications whenever qualifying changes in a Web page have been detected, regardless of whether the modifications reflect the latest version of the file or not"), 38 ("Williams is both anticipating and accommodating situations in which computer E2 may store a different, but not necessarily a latest, version of a file than computer E1"). Patent Owner asserts inappropriate file updating is avoided in claims 37 and 42 "by determining whether the second computer has a latest version of a file and, in this way, establishes that the latest version of the file will not be overwritten by changes to a file that occurred previously to the changes of the latest version of the base file simply because the files are not the same (i.e., one of the files has been modified)." PO Resp. 8. Patent Owner cites to Dr. Mohaptra's Declaration[8], which uses the exact same language and provides no further explanation as to why the determining limitation should be construed to prevent the inappropriate file updating that is allegedly an issue only in the prior art. Ex. 2009 ¶ 17.

Petitioners explain that there is nothing in the'799 Patent that prevents a copy of a subject file from being updated on a client, but that an ordinarily skilled artisan would have understood that the system was not intended to operate in that way. Hr'g Tr. 7:23–8:10. Petitioners further argue that claims 37 and 42 recite the determining is done "without interaction with the client[/second computer]." *Id.* at 8:16–18. Moreover, Petitioners explain that the '799 Patent describes the process of the determining limitation

[8] Dr. Prasant Mohaptra is Patent Owner's declarant, whose Declaration was submitted as Exhibit 2009.

assuming that the system is operated normally, and that the server can use traditional mechanisms (e.g., comparing time stamps) to determine whether the second computer has a current version of the file. *Id.* at 8:16–24. Thus, Petitioners implicitly assert that a person having ordinary skill in the art would not have understood the determining limitation, which requires the determining to be done *without interaction with the second computer*, to require an absolute certainty that the second computer have the latest version. *Id.* at 8:16–13:20. Rather, Petitioners implicitly assert that the proper construction of the determining limitation is that the first computer, assuming normal operation of the system and using techniques known to those of ordinary skill in the art, detects whether there is a difference between a current version and an old version of the file at the first computer, and updates the second computer if there is a difference. *Id.*

With respect to the latest version, a review of the Specification of the ’799 Patent reveals only one occurrence of the term “latest.” Ex. 1001, 13:12–14 (“For the next file change, the server will take yet another snapshot and compare it against the latest snapshot and so on.”). The ’799 Patent does, however, consistently refer to updating an old version of a file to a current version of a file.

Although Patent Owner asserts that claims 37 and 42 would not allow a newer file to be overwritten by an older file (PO Resp. 7–8), Patent Owner does not point to evidence sufficient to support such a restriction. Patent Owner does not explain how the recited language requires an implementation that prevents issues with transit delays, tampering, or other uncertainties with respect to the first computer’s most recent version not being as recent as another version of the file. Patent Owner simply points to the recited language as the explanation for why claims 37 and 42 do not suffer from the same problems as the prior art.

The closest Patent Owner comes to providing support for its proposed construction is its argument, without citation to supporting evidence, that claims 37 and 42 use the timestamp of when a change was made (rather than when the server received the change) to determine the latest version. Hr'g Tr. 29:15–31:13. Patent Owner explains that, even when using the timestamp of when a change was made, the server relies "upon a comparison of its *own* time stamps" because "the server does not have the time stamps for the client computer." Hr'g Tr. at 19:12–17 (emphasis added). Nevertheless, even accepting the argument that the server uses the timestamp of when a change was made, as opposed to received, by the server, there are two issues. First, in the example provided by Patent Owner, the first computer may receive a second update (where the change was made later) prior to receiving a first update (where the change was made earlier) due to network latency. *Id.* at 30:4–9, 30:21–31:9. There is no disclosure in the '799 Patent indicating that an evaluation made by the first computer, in the time between receiving the first and second updates, would result in a determination that the second update was the latest version. This is inconsistent with Patent Owner's position that the server is always using the latest change to provide an update. Second, regardless of when updates are made, there is no recited restriction on how the first computer determines what version is the current or latest version.

There are various references throughout the ’799 Patent that discuss the first computer, or server, determining whether an update file needs to be sent to the second, or client, computer(s). *See, e.g.*, Ex. 1001, Abstract (“The server periodically monitors the subscription file to determine if it has been altered before generating an update file”), 4:32–34 (“The update file is only generated when the server computer determines that the subscription file has changed . . . [t]he user determines the periodicity of the checks to determine if the file has been altered”), 6:51–55 (“the user then determines the polling or monitoring interval for the server to check for changes and also what to do when changes occur, i.e., package and send file changes or simple notification”). The ’799 Patent also discloses that the server periodically monitors files or folders to which clients subscribe for changes in order to determine whether an update needs to be generated. *See, e.g., id.* at 3:41–44, 4:32–39, 7:56–57, 9:28–31. One example provided in the ’799 Patent is that the server “polls files or subfolders at either [sic] user-defined intervals for any changes to date, time stamps.” *Id.* at 6:59–60. Each of these descriptions explains that the server (the first computer) makes a determination without comparing the subject file (or its timestamp) to the version of the file stored at the client (the second computer) (or its timestamp).

In light of the considerations discussed above, we do not see anything that would lead us to conclude that the proper construction of the determining limitation requires absolutely certainty. From the perspective of the first computer, the latest version, as recited in claims 37 and 42, refers to the current version accessible by the first computer, as determined by the first computer at the time a check is made. Thus, at the point in time when the first computer makes its determination, the first computer merely needs to determine whether it has already sent its current version to the second computer. Considering all of the above, an ordinarily skilled artisan would have understood the plain and ordinary meaning of the "determining limitation" to be determining, by the first computer using known techniques based on information it has access to and without interaction with the second computer, whether the last version of the subject file sent to the second computer is not the same as the current version of the subject file.

B. Submitted Evidence

1. Williams (Ex. 1006)

Williams describes a fine-grained incremental backup system and process. Ex. 1006, 19:26–22:14. Figure 25 of Williams is reproduced below:

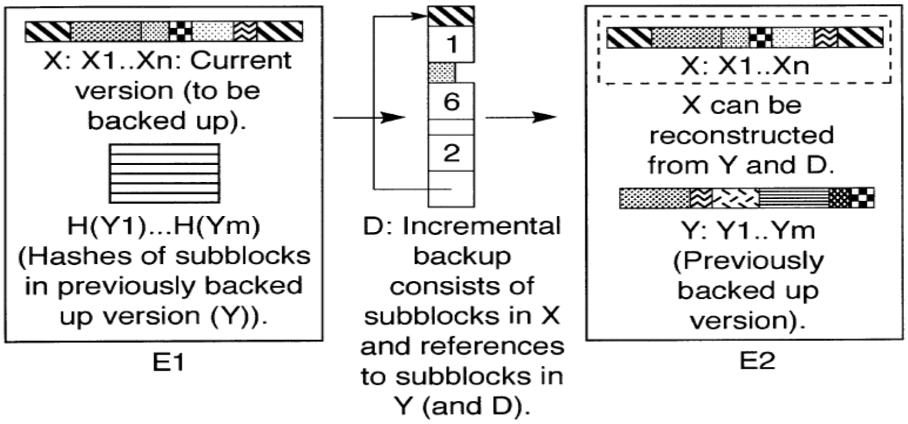
X: X1..Xn: Current version (to be backed up).
H(Y1)...H(Ym) (Hashes of subblocks in previously backed up version (Y)).
E1
1
6
2
D: Incremental backup consists of subblocks in X and references to subblocks in Y (and D).
X: X1..Xn
X can be reconstructed from Y and D.
Y: Y1..Ym (Previously backed up version).
E2

Figure 25 of Williams illustrates the backup process for two network computers.

Williams describes that a file on computer E1 is backed up to computer E2, such that both E1 and E2 have a copy of the same version of the file, referred to as Y. When the file is modified, computer E1 now has a new (current) version of the file, referred to as X. As shown in Figure 25 of Williams, at that time each of the network computers (E1 and E2) has a version of the same file (X and Y, respectively). Williams then determines the information to send from computer E1 to E2 necessary for E2 to generate a copy of X and sends that incremental backup information in a file, referred to as D, from computer E1 to computer E2. Computer E2 subsequently may generate a copy of the current version of the file, X, by using its prior version of the file, Y, and the incremental backup information file, D. Ex. 1006, 19:29–34, 19:63–20:2.

For further improvement, Williams indicates that copies of the previous versions of the file system should be retained. Ex. 1006, 21:62–65. This means that computer E2 should maintain both versions of the file, Y (the previous version) and X (the current version) when generating the current version. *Id.* Therefore, computer E2 eventually may store every prior version of that file.

As explained in Williams, computer E1 compares the hash of the old version of the file, Y, against the hash of current version of the file, X, to determine whether the file has changed. Ex. 1006, 19:44–46. If the file has changed, computer E1 partitions the current version of the file into subblocks, and compares the hashes of these subblocks with the hashes of previous version of the file that are stored in shadow file S of computer E1, to find all identical hashes. Ex. 1006, 19:48–51. "Identical hashes identify identical subblocks in [version] Y that can be transmitted by reference." Ex. 1006, 19:51–52. Computer E1 then transmits the incremental backup file D as a mixture of raw subblocks and references to subblocks whose hashes appear in the shadow file S and which are known to appear as subblocks in the prior version of the file, Y. Ex. 1006, 19:52–55.

To reconstruct a duplicate of the current version of the file, X, from the prior version of the file, Y, and incremental backup file, D, computer E2 partitions Y into subblocks and calculates the hashes of subblocks. Ex. 1006, 19:66–20:1. "It then processes the incremental backup information, copying subblocks that were transmitted raw and looking up the references" in Y. Ex. 1006, 20:2–5.

2. Miller (Ex. 1004)

Miller describes a method and system for using small difference ("diff") files to update or revise large computer files. Ex. 1004, 1:11–13. The diff files are small files that indicate the differences between the large computer files and preexisting computer files. *Id.* at 1:13–15. Figure 1 of Miller, reproduced below, illustrates how the process works.

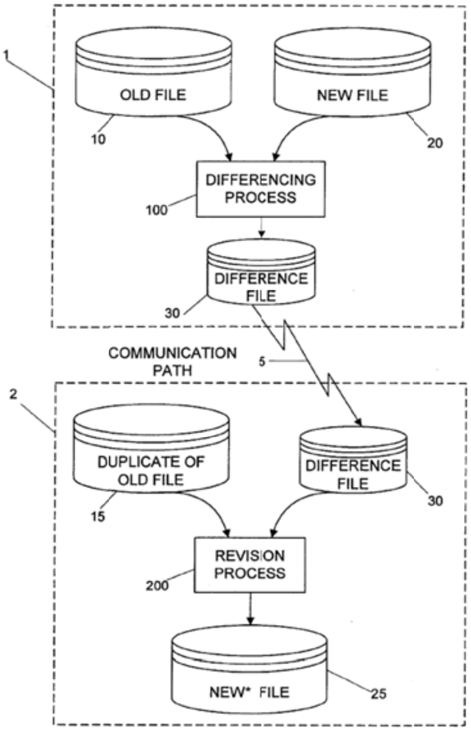

Figure 1 of Miller illustrates the process of generating a diff file at a first computer system by identifying differences between an old file and a new file, communicating the diff file to the second computer system over a communication path, and generating a copy of the new file at a second location using the diff file and a copy of the old file present at the second location.

Miller's diff file indicates changes between the old file and the new file using a minimal number of bytes. *Id.* at 2:21–24, 31–33. Miller's diff file is comprised of copy and insert commands. *Id.* at 13:24–30. Miller's first computer system then communicates the diff file to the second computer system, using any means for communicating between computer systems, including e-mail. *Id.* at 5:10–17. The second computer system then generates a new file, either when invoked explicitly or through receipt of a self-extracting execution file, copying segments from its copy of the old file and segments from the diff file. *Id.* at 15:25–43.

3. Balcha (Ex. 1003)

Balcha discloses a method for synchronization of files. Ex. 1003, 1:5–7. In particular, a synchronized file exists on two different servers, and changes made to one file must be reflected in the other file. Ex. 1003, 1:42–44. Figure 1 of Balcha, reproduced below, illustrates a computer network with two servers using file synchronization.

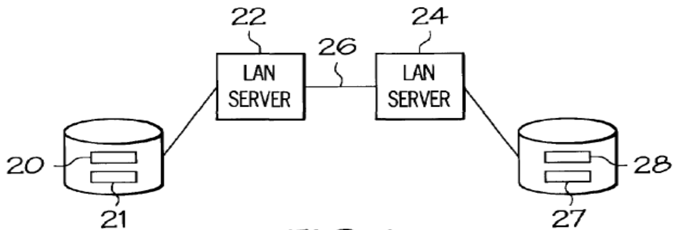

FIG. 1

As shown in Figure 1 of Balcha, servers (22 & 24) are interconnected via a network 26, and each server (22 & 24) maintains a copy of a base file (21 & 27) and a base signature file (20 & 28). Ex. 1003, 4:51–53. The base files (21 & 27) should be identical, but either base file can be modified at either server. Ex. 1003, 4:53–61. *Upon detection of a modification to the file*, the detecting server (e.g., server 22), uses the respective base signature file (e.g., base signature file 20) to generate a new delta file, and communicates the delta file over network 26 to server 24. Ex. 1003, 4:61–66 (emphasis added). Server 24 uses the delta file to update the base file 27, and recalculates the base signature file 28. Ex. 1003, 4:66–67. As a consequence, the base files on the servers will stay in synchronization with minimal transfer of data over network 26. Ex. 1003, 5:1–3.

Figure 3 of Balcha is reproduced below:

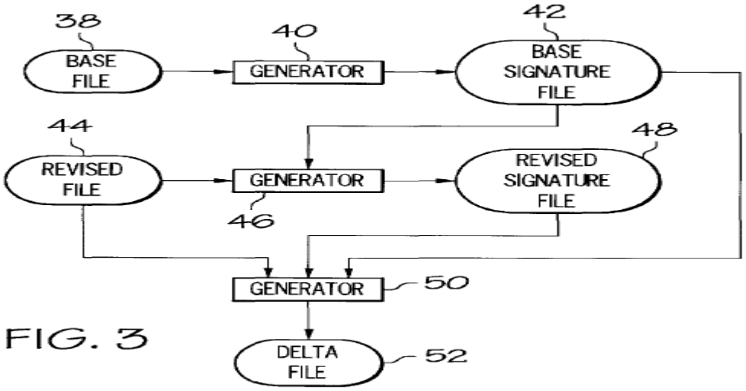

Figure 3 of Balcha illustrates the relationship of the files.

Referring to Figure 3 of Balcha, the base signature file (42) contains a plurality of cyclic redundancy check (CRC) values derived from the data contained in the base file (38). Ex. 1003, 3:1–3, 3:21–28, 7:46–49. When a revised version of the base file (44) is created, a revised signature file (48), including a plurality of revised bit patterns, is generated from the revised file

(44). Ex. 1003, 3:4–6, 7:49–53. "*Each revised bit pattern is compared to the base bit patterns in base signature file 42*." Ex. 1003, 7:57–59 (emphasis added). "For each revised bit pattern that matches a base bit pattern in base signature file 42, it is stored in revised signature file 48, along with an offset indicating the location in revised file 44 of the beginning of the block of data represented by the revised bit pattern." Ex. 1003, 7:59–63.

Based on the differences between the base signature file and the revised signature file, a delta file reflecting the differences between the base file and the revised file is generated. Ex. 1003, 3:7–10, 3:50–54. The delta file contains primitives, such as insert, modify, and delete primitives, which are commands that can be applied to a previous version of the file to generate the revised file. Ex. 1003, 3:54–58.

4. Freivald (Ex. 1005)

Freivald describes a change-detection web server that automatically checks pages for changes. Ex. 1005, Abstract. Freivald discloses a subscription service in which users that wish to be notified of changes to Web pages or portions thereof may register those pages. *Id.* at 7:3–15. An associated subscription server, referred to as a minder, periodically checks to see if the registered pages have changed and, upon detecting changes, notifies the user by sending an email. *Id.* at 5:18–43.

5. Update/Synchronization Schemes

Both parties agree that retain-by-default and discard-by-default are the two schemes used in file synchronization or file backup. *See, e.g.,* Pet. 51; Ex. 1007 ¶¶ 22–27, 46–48; Hr'g Tr. 17:11–18:4. Briefly, retain-by-default indicates that, when generating a new version of the file, each portion of the old version of the file is kept unless the update explicitly indicates removing that portion. Conversely, discard-by-default indicates that, when generating a new version of a file, each portion of the old version of the file is discarded unless the update explicitly indicates keeping that portion. *See* Ex. 1007, ¶¶ 22–27.

Although the specific commands inserted into an update file by the first computer differ, the operations executed by the second computer upon receipt of the update file are the same. *Id.* ¶¶ 22, 26, 46. Put another way, the schemes merely differ in whether the identified blocks are those to be kept or those to be discarded, whereas the actual instructions executed at the second computer depend on the goal that the second computer is programmed to accomplish, not the commands inserted into the update file. *Id.*

In particular, if the second computer is programmed to generate a new backup file upon receipt of an incremental update file (retaining the prior version backup file), the second computer will interpret the insert instructions in the update file to copy the new sections from the update file into the new backup file. *Id.* ¶¶ 25, 37. If the update file uses copy commands (discard-by-default scheme), the second computer will interpret copy commands in the update file such that the second computer will copy the identified sections from the prior version of the backup file, ignoring the sections not identified as needing to be copied. *Id.* ¶ 57. Similarly, if the update file uses delete commands (retain-by-default scheme), the second computer will interpret the delete instructions to ignore the sections identified, thus retaining sections not identified to be deleted from the prior version backup file by copying them from the prior version to the version being generated. *Id.* ¶¶ 35–36. Therefore, generating update files using either a retain-by-default scheme or a discard-by-default scheme may meet the recited "writing a command . . . to copy" because either a copy command or a delete command may cause the second computer to duplicate information or data from an earlier version of a file into a current version of a file.

C. Analysis of Whether Claims 1, 12, 23, 24, 30, 31, 37, and 42 Are Unpatentable As Anticipated by Williams Under 35 U.S.C. § 102(e)

The Board instituted trial on Petitioners' challenge of anticipation of claims 1, 12, 23, 24, 30, 31, 37, and 42 by Williams under 35 U.S.C. § 102(e). Dec. on Inst. 17–20, 29. As described above, one aspect of Williams is directed to a fine-grained incremental backup system. Ex. 1006, 19:26–22:14. Petitioners allege that each element of claims 1, 12, 23, 24, 30, 31, 37, and 42 is disclosed by Williams and provide detailed analysis and claim charts mapping the limitations of the claims to disclosures within Williams, including mapping the incremental backup file D to the recited update, computer E1 to the recited first computer, computer E2 to the recited second computer, the file's subblocks to the recited segments, and the shadow file to the recited signature list. Pet. 32–40. Moreover, as argued by Petitioners, Williams discloses placing commands in the incremental backup file, or the update, which allow computer E2 to generate the new version of the file, X, from the old version of the file, Y, and the incremental backup file, D. *Id.*; Ex. 1006, 19:45–20:5. Williams also discloses that no incremental backup file needs to be sent to computer E2 if the file has not changed[9], which can be determined by comparing the current file to the shadow file or by evaluating "the modification date file attribute of the file." Ex. 1006, 19:45–49; Pet. 38–39.

1. Claims 1, 12, 23, 24, and 30

Patent Owner argues claims 1, 12, 23, 24, 30, and 31 are not anticipated because "Williams does not teach a command to copy or a command to insert as recited in independent claims 1, 12, 23, and 30 and construed by the Board." PO Resp. 26 (emphasis omitted). In particular, Patent Owner asserts that the incremental backup file includes raw subblocks not present in the original version of the file and references to subblocks that were present in the original version of the file and should be retained, but that no "command to copy (or it[s] equivalent)" is present in the incremental backup file. *Id.* at 27–28. Patent Owner argues Williams does not provide any other information regarding the references to the new subblocks and that those references to new subblocks are not commands to copy. Hr'g Tr. 17:7–10. Patent Owner further asserts that Williams merely updates a prior version of a file to become a copy of the current version of the file, so no action of copying would result from any instructions or data in the incremental backup file. *Id.* at 28. Patent Owner argues the distinction between the claims at issue and Williams's disclosure focuses on "the

[9] Patent Owner asserted that Williams's update process "takes place without regard to whether or not the other computer has the latest version of the file." Hr'g Tr. 20:9–14. To the extent Patent Owner is asserting that Williams backs up a file even if it has not changed, that is incorrect. Ex. 1006, 19:47–49 ("If X [on computer E1] hasn't changed, [and X and Y are still the same], there is no need to perform any further backup action.")

operations that result from the application of [the increment backup] file as taught by Williams," not the literal language of an instruction in Williams's incremental backup file. *Id.* at 29.

We find Patent Owner's argument unavailing. Petitioners pointed to portions of Williams that disclose the second computer generating a copy of the current version of the file, X, using the old version of the file, Y, stored at the second computer, and the incremental backup file, D, received from the first computer. Pet. 33, 35–36 (citing Ex. 1006, 19:66–20:9, Fig. 25).

Notwithstanding Patent Owner's argument that the distinction focuses on the resultant operations rather than the language of the instruction in the backup file, Patent Owner implies that only a discard-by-default scheme can disclose the "writing a command . . . to copy" limitation. *See* Hr'g Tr. 17:7–18:25. Patent Owner reiterates that Williams merely indicates that its incremental backup file includes "references to file [segments], and doesn't explain the references." *Id.* at 18:15–17. Patent Owner argues that, because Williams is equally likely to implement a retain-by-default scheme or a discard-by-default scheme, Williams does not disclose inserting a command to copy, either explicitly or inherently. *Id.* at 18:17–25.

First, Patent Owner's argument that each scheme is equally likely is not persuasive. Petitioners point to Williams's disclosure that the incremental backup file sends references to blocks in the old version, Y, that match blocks in the current version, X and, therefore, should be included when generating a copy of the current version, X, at the second computer. Pet. 36 (citing Ex. 1006, 21:5–11). Identifying blocks that are to be kept (as opposed to identifying blocks to be discarded) is a characteristic of a discard-by-default scheme, nullifying Patent Owner's argument that Williams just as likely could be using a retain-by-default scheme.

Nevertheless, as discussed above, the scheme used does not inform us whether the command inserted into the update, or incremental backup file, causes the second computer to duplicate data or information from the old version of the file into the second computer's copy of the current version of the file, as required by the "writing a command . . . to copy" limitation. In order to determine that, we need to look at what the second computer does with the incremental backup file and, in particular, whether the second computer copies data segments from an old version of the file to a copy of the current version of the file. Patent Owner acknowledges that "the semantics are important" with respect to whether a new file is generated for the copy of the current version of the file at the second computer or whether the old version of the file is merely replaced with the new version. Hr'g Tr. 27:10–16.

We look to the relevant portions of Williams cited by both parties to understand what Williams's second computer does with the incremental backup file. Williams "processes the incremental backup information, copying subblocks that were transmitted raw and looking up the references either in Y or in the part of X already reconstructed." Ex. 1006, 20:2–5. Contrary to Patent Owner's argument that the new version merely replaces the old version, that description in Williams provides some evidence that Y and X are separately co-existing files at the second computer. Moreover, as addressed in the discussion of Williams above, copies of previous versions may be retained, such that computer E2 may maintain both its copies of old versions of the file, e.g., Y_1–Y_n, and its copy of the current version of the file, X. Ex. 1006, 21:62–67. The evidence therefore supports the finding that Williams's second computer, E2, does duplicate data or information as a result of the commands Williams's first computer, E1, inserted into the incremental backup file, D, meeting the recited "writing a command . . . to copy."

Patent Owner also argues "[t]he copying that [Petitioners] referred to was a copying of information into the update file, but that is not the copying referenced in the claim language. The claim language represents copying occurring at the second computer." Hr'g. Tr. 16:14–17. It is unclear to which of Petitioners' arguments Patent Owner refers. To the extent Patent Owner argues the portions of Dr. Mohaptra's deposition testimony cited by Petitioners regarding how Williams's copying works (*see* Ex. 1021, 3–5 (quoting Ex. 1019, 42:21–25, 44:5–8, 44:20–45:8) refer to copying information into the update file, that argument is not persuasive. For example, Dr. Mohaptra agreed that both blocks from the old version of the file, Y, on computer E2 and blocks from the incremental update file, D, are copied into X. Ex. 1019, 41:8–45:8. There is no ambiguity in Williams that only computer E1 generates the update file and only computer E2 generates a version of the file using blocks from the update and from an old version of the file. *See* Ex. 1006, 19:27–22:14.

We have reviewed the Petition, the Patent Owner Response, and Petitioners' Reply, as well as the relevant evidence discussed in those papers. We are persuaded, by a preponderance of the evidence, that claims 1, 12, 23, 24, 30, and 31 are anticipated by Williams under 35 U.S.C. § 102.

2. *Claims 37 and 42*

Patent Owner argues claims 37 and 42 are not anticipated because "Williams does not teach determining whether the second computer has a latest version of a file and generating an update if the second computer does not have a latest version of a file." PO Resp. 38. In particular, Patent Owner asserts Williams "generates the update when the backup system determines that a backup should be made," and that "computer E1 cannot be certain whether or not computer E2 stores a latest version of the subject file." *Id.* Patent Owner further argues that "Williams is a back-up system, and the back-up operates according to a schedule . . . whether or not it's the latest version," and that Williams does not discuss time stamps. Hr'g Tr. 20:1–8.

As explained above, our construction of the determining limitation does not require absolute certainty that the current version is the latest version. Moreover, contrary to Patent Owner's arguments, Petitioners point out that Williams discloses determining whether a file needs to be updated before executing the backup procedure and further discloses using time stamps to determine whether the second computer has the current version. Hr'g Tr. 40:7–20, 42:15–43:7 (citing Ex. 1006, 19:47–49, 22:10–14).

We have reviewed the Petition, the Patent Owner Response, and Petitioners' Reply, as well as the relevant evidence discussed in those papers. We are persuaded, by a preponderance of the evidence, that claims 37 and 42 are anticipated by Williams under 35 U.S.C. § 102.

D. Analysis of Whether Claims 5–10 and 16–21 Are Unpatentable as Obvious Over Williams and Miller Under 35 U.S.C. § 103(a)

The Board instituted trial on Petitioners' challenge of obviousness of claims 5–10, which depend ultimately from claim 1, and 16–21, which depend ultimately from claim 12, over Williams and Miller. Dec. on Inst. 20, 30. Claims 16–21 are similar in substance to claims 5–10, but claims 16–21 recite computer readable media. Petitioners allege that each additional element of claims 5–10 and 16–21 is taught by the combination of Williams and Miller, provide detailed analysis and claim charts mapping the limitations of the claims to disclosures of Williams and Miller, and provide reasons for combining Williams and Miller. Pet. 41–45. Patent Owner argues only that claims 5–10 and 16–21 are patentable for the same reasons argued with respect to claims 1 and 12, which we addressed above.

Claims 5 and 16 recite an additional limitation relating to sending the update as an executable attachment in an e-mail. Petitioners allege that the additional limitation in claims 5 and 16 is taught by Miller, which "discloses a system, method, and file structure for transmitting a difference file from a first computer to a second computer, *as an executable file,* which allows the second computer to create a copy of the most up-to-date version of a monitored file using the second computer[']s copy of an old version of the monitored file and the executable difference file." Pet. 42–43 (emphasis added) (citing Ex. 1004, 2:38–46, 5:9–16, 5:35–39, and 15:41–43). Petitioners argue combining Williams and Miller is the mere substitution of one known method for another with predictable results, and that such a combination would have been obvious because Williams does not restrict the method of communication, and using an e-mail would have been a known method of communication. Pet. 41–42.

We have reviewed the Petition, the Patent Owner Response, and Petitioners' Reply, as well as the relevant evidence discussed in those papers. We are persuaded, by a preponderance of the evidence, that claims 5–10 and 16–21 are obvious in view of the combination of Williams and Miller under 35 U.S.C. § 103.

E. Analysis of Whether Claims 37 and 42 Are Unpatentable As Anticipated by Balcha Under 35 U.S.C. § 102(e)

The Board instituted trial on Petitioners' challenge of anticipation of claims 37 and 42 by Balcha under 35 U.S.C. § 102(e). Dec. on Inst. 23–24, 30. As described above, Balcha is directed to file synchronization. *See, e.g.*, Ex. 1003, 1:5–7, 1:42–44. Petitioners allege that each element of claims 37 and 42 is disclosed by Balcha and provide detailed analysis and claim charts mapping the limitations of the claims to disclosures within Balcha, including mapping Balcha's delta file to the update, Balcha's detecting server to the recited first computer, Balcha's server that receives the delta file to the recited second computer, Balcha's base file to the recited file, Balcha's determination of whether a base file has been revised by using its own base signature file to the recited determining limitation, and Balcha's communication of the delta file to the receiving server to the recited transmission of the update. Pet. 26–28.

Patent Owner argues Balcha does not disclose the determining limitation because Balcha "discloses detecting a modification to the file without regard to whether the modified file is, indeed, the latest version of the file" and "the detection of the modification of base file 21 is in no way related to, or dependent upon, a determination that base file 21 is the latest version of the base file." PO Resp. 7.

Based on our construction of the determining limitation, particularly in light of the fact that the determining limitation recites that the determination is made by the first computer without interaction with the second computer, we do not agree with Patent Owner that Balcha fails to disclose the determining limitation.

We have reviewed the Petition, the Patent Owner Response, and Petitioners' Reply, as well as the relevant evidence discussed in those papers. We are persuaded, by a preponderance of the evidence, that claims 37 and 42 are anticipated by Balcha under 35 U.S.C. § 102.

F. Analysis of Whether Claims 37 and 42 Are Unpatentable as Obvious Over Balcha and Freivald Under 35 U.S.C. § 103(a)

The Board instituted trial on Petitioners' challenge of obviousness of claims 37 and 42 over Balcha and Freivald. Dec. on Inst. 28–30. Petitioners allege that each element of claims 37 and 42 is taught by Balcha, and that Freivald provides a further teaching with respect to determining whether a second computer has a latest version. Petitioners provide detailed analysis and claim charts mapping the limitations of the claims to disclosures of Balcha and Freivald, and provide reasons for combining Balcha and Freivald. Pet. 45–47, 54–57. Patent Owner argues that neither Balcha nor Freivald teaches the determining limitation recited in claims 37 and 42. Balcha's teachings with respect to the determining limitation were addressed above.

Specifically, Petitioners assert that Freivald discloses a minder that, "without interaction with a client computer, determines whether the client computer has the most recent version of a web page document and, if the client's version is out of date, transmits an updated copy of the document to the client." *Id.* at 55–56 (citing Ex. 1005, 6:61–67, 7:25–39). Petitioners also argue Freivald discloses comparing revision dates or time stamps to determine whether the second computer has a latest version of the file, in the same way as described in the '799 Patent. *Id.* at 56 (citing Ex. 1001, 6:57–63, Ex. 1005, 2:31–36). Petitioners argue a skilled artisan would have been motivated to combine Balcha and Freivald because they both serve the purpose of keeping remote versions of files up-to-date, and Freivald's change-detection server would have been a convenient way to detect changes in Balcha's base files. *Id.* at 46–47. Petitioners further argue one would have combined Balcha and Freivald because Freivald's change-detection method was one of a limited number of known ways to detect whether a file had been modified such that an update would need to be distributed, the use of which would have been a known solution providing a predictable result. *Id.* at 47.

We have reviewed the Petition, the Patent Owner Response, and Petitioners' Reply, as well as the relevant evidence discussed in those papers. As discussed above, we are persuaded that Balcha discloses each of the limitations for which it is relied on. Moreover, we are persuaded, by a preponderance of the evidence, that Balcha and Freivald is a proper combination, that Freivald provides a further teaching of a first computer determining whether a second computer has a latest version of a file and, therefore, that claims 37 and 42 are obvious in view of the combination of Balcha and Freivald under 35 U.S.C. § 103.

G. Analysis of Whether Claims 1, 5, 9, 10, 12, 16, 20, 21, 23, 24, 30, and 31 Are Unpatentable Obvious Over Balcha and Miller Under 35 U.S.C. § 103(a)

The Board instituted trial on Petitioners' challenge of obviousness of claims 1, 5, 9, 10, 12, 16, 20, 21, 23, 24, 30, and 31 over Balcha and Miller. Dec. on Inst. 24–26, 30. Claims 12, 16, 20, 21, 30, and 31 differ from claims 1, 5, 9, 10, 23, and 24, respectively, only in that claims 12, 16, 20, 21, 30, and 31 recite computer readable media rather than methods. Petitioners allege that each element of claims 1, 5, 9, 10, 12, 16, 20, 21, 23, 24, 30, and 31 is taught by the combination of Balcha and Miller, provide detailed analysis and claim charts mapping the limitations of the claims to disclosures of Balcha and Miller, and provide reasons for combining Balcha and Miller. Pet. 16–25.

Specifically, with respect to claims 1, 12, 23, 24, 30, and 31, Petitioners assert that Balcha discloses each limitation and that Miller further discloses inserting a copy command into a difference file when it finds matching segments. *Id.* at 21–22 (citing Ex. 1004, 5:52–56), 25–26. Petitioners also argue Miller discloses each additional limitation recited in dependent claims 5, 9, 10, 16, 20, and 21. *Id.* at 23–26. In particular, Petitioners map Miller's self-extracting execution file sent in an electronic mail message that generates a copy at the second computer to "transmitting the update to the second computer as an executable attachment . . .," recited in claims 5 and 16. *Id.* at 23–24, 26 (citing Ex. 1004, 5:9–16, 5:35–39, 15:41–43). Petitioners also point to specific sections of Miller that disclose updating either software or a document as recited in claims 9 and 20 and claims 10 and 21, respectively. *Id.* at 24–26 (citing Ex. 1004, 1:42–44, 2:21–24, 5:21–26). Petitioners argue a skilled artisan would have been motivated to combine Balcha and Miller because they both use delta files to allow remote computers to update versions of a file, and because updating software, as disclosed by Miller, in place of Balcha's data files is just the substitution of one well-known file type for another with predictable results. *Id.* at 17. Finally, Petitioners argue that copying is one of a limited number of options for updating an old version to a new version and that it was within the skill level of an ordinarily skilled artisan to elect to use the copy mechanism. Hr'g Tr. 45–46.

Patent Owner argues that Balcha does not teach a command to copy because it uses only insert, modify and delete primitives, and that the data segments that are the same between versions merely remain in the new version of the file, without needing to copy those segments. PO Resp. 12–13. Patent Owner further argues that Balcha, therefore, has no need for a copy primitive so Balcha cannot teach or suggest writing a command to copy data. *Id.* at 13. Patent Owner admits that the combination of Balcha and Miller teaches each of the recited limitations, but argues one would not combine Balcha with Miller because Miller stresses making the diff files as small as possible, and adding a copy command to Balcha's system would unduly increase the size of the diff file. *Id.* at 19–20; Hr'g Tr. 25, 33 ("if you had the Balcha and the Miller combination and you found that it was a proper combination, then the elements of the claims are there").

In light of our construction of "writing a command . . . to copy," at least Miller teaches "writing a command . . . to copy." Specifically, regardless of the name of the command ("copy" versus "delete") used, Miller discusses creating a new file and, using the diff file and the old file, "[s]trings will be copied or inserted into the new file." Ex. 1004, 15:62–63. Moreover, although Balcha is not clear regarding whether the second computer generates a new file or merely updates the old file, Petitioners' argument that there are only limited options, and electing to generate a new file (thus retaining prior versions) would have been obvious for a skilled artisan to try, is persuasive.

We find persuasive the various reasons presented by Petitioners for combining Balcha and Miller, which are both systems and methods directed to updating files on a remote computer. As pointed out by Petitioners, there were a limited number of options for executing incremental backup (or diff) files.

We have reviewed the Petition, the Patent Owner Response, and Petitioners' Reply, as well as the relevant evidence discussed in those papers. We are persuaded, by a preponderance of the evidence, that the combination of Balcha and Miller teaches each of the limitations and that Balcha and Miller is a proper combination, and therefore, that claims 1, 5, 9, 10, 12, 16, 20, 21, 23, 24, 30, and 31 are obvious in view of the combination of Balcha and Miller under 35 U.S.C. § 103.

H. Analysis of Whether Claims 6–8 and 17–19 Are Unpatentable Obvious Over Balcha, Miller, and Freivald Under 35 U.S.C. § 103(a)

The Board instituted trial on Petitioners' challenge of obviousness of claims 6–8, which depend from claim 5, and 17–19, which depend from claim 16, over Balcha, Miller, and Freivald. Dec. on Inst. 24–26, 30. Claims 17–19 differ from claims 6–8 only in that claims 17–19 recite computer readable media rather than methods. Patent Owner argues only that claims 6–8 and 17–19 are patentable for the same reasons argued with respect to the patentability of claims 1 and 12 over Balcha and Miller, which we addressed above.

Petitioners allege that each additional limitation recited in claims 6–8 and 17–19 is taught by Freivald, provide detailed analysis and claim charts mapping the limitations of the claims to disclosures of Balcha, Miller and Freivald, and provide reasons for combining Freivald with the Balcha-Miller system. Pet. 28–31. Specifically, claims 6 and 17 recite checking whether the version of the file was altered prior to executing the steps necessary to generate the update and executing those steps only if a change is detected. Claims 7 and 18 recite "performing a check is performed at periodic intervals," and claims 8 and 19 recite performing a check is done by comparing current and earlier time stamps.

Petitioners argue a skilled artisan would have been motivated to combine Freivald with Balcha-Miller because all of the prior art references relate to sending updates to remote computers to allow the remote computer to amend files, and because incorporating Freivald's teachings related to determining whether a file has been altered is merely substituting well-known methods for those disclosed in Balcha and Miller, leading to predictable results. *Id.* at 29. Moreover, Petitioners argue there were a limited number of options "to determine when files should be updated (e.g., periodic polling, waiting to receive an update request)" and that each option would lead to a predictable and known solution. *Id.*

We have reviewed the Petition, the Patent Owner Response, and Petitioners' Reply, as well as the relevant evidence discussed in those papers. We are persuaded, by a preponderance of the evidence, that Balcha, Miller, and Freivald is a proper combination, that the combination teaches the subject matter of claims 6–8 and 17–19, and, therefore, that claims 6–8 and 17–19 are obvious in view of the combination of Balcha, Miller, and Freivald under 35 U.S.C. § 103.

I. Motion To Amend

Patent Owner moves to substitute claim 47 for claim 42, if we determine claim 42 is unpatentable. Mot. to Amend 1. Because we determine that Petitioners have demonstrated, by a preponderance of the evidence, that all challenged claims are unpatentable, including claim 42, Patent Owner's Contingent Motion to Amend is before us for consideration. Proposed substitute claim 47 is an independent claim, and is reproduced below.

> 47. A computer readable storage medium, comprising computer readable program code embodied on said computer readable storage medium, said computer readable program code for programming a first computer to provide updates for transmission to a second computer that permits the second computer to obtain most recent versions of files, the computer readable program code causing the first computer to perform the following steps:
>
> (a) determining whether the second computer has a latest version of a file, wherein said determining is performed by the first computer without interaction with the second computer <u>by comparing representations of segments of</u>

> the latest version of the file with representations of segments of an earlier version of the file in which ends of each of the segments of the earlier version of the file are defined by segment delimiters that are statistically determined to be optimal division points for the segments;
>
> (b) generating an update, if the second computer does not have a latest version of the file, wherein said generating is performed by the first computer without interaction with the second computer; and
>
> (c) transmitting the update from the first computer to the second computer.

Mot. to Amend 1–2 (emphasis added by Patent Owner to indicate changes).

A motion to amend is a motion under 37 C.F.R. § 42.20, and is subject to the requirements of that rule. The rule includes that "[t]he moving party has the burden of proof to establish that it is entitled to the requested relief." 37 C.F.R. § 42.20(c). Thus, the Patent Owner as movant bears the burden to demonstrate patentability and compliance with 37 C.F.R. § 42.221.

This burden may not be met merely by showing that the proposed claims are distinguished over the prior art references applied to the original patent claims. Instead, Patent Owner must show that the proposed substitute claims are patentable over the prior art in general. *See Idle Free Sys., Inc. v. Bergstrom, Inc.*, Case IPR2012–00027, slip op. at 33 (PTAB Jan. 7, 2014) (Paper 66). "An *inter partes* review is neither a patent examination proceeding nor a patent reexamination proceeding." *Id.* If a motion to amend is granted, the proposed substitute claims will be added directly to the patent, without examination. *Id.* Petitioners cannot be relied on to fill the role of an Examiner because their interests and motivation are not necessarily aligned with that of the general public. For the reasons discussed below, Patent Owner's Motion to Amend is denied.

Patent Owner acknowledges[10] that "[t]ransmitting files between computers as a number of different segments (rather than as an entire file) has been a technique used for many decades." Mot. to Amend 12; *see also* Hr'g Tr. 34. Patent Owner further explains that, at the time of invention of the '799 Patent, files were segmented using either fixed length segments or variable length segments, and each segment was identified by delimiters. Mot. to Amend 12–13. Patent Owner also acknowledges that prior art systems, including Williams, taught methods of file synchronization using variable length segments. *Id.* at 13. Patent Owner explains that there were various ways of determining the boundaries for variable length segments, such as denoting boundaries when a certain character or set of characters was present in a file. *Id.* at 13–14. Patent Owner even acknowledges that "Williams also recognized that it may be desirable to employ different file segmenting strategies in a single application." *Id.* at 14.

Patent Owner argues that the specific method of segmentation recited in proposed claim 47 is missing in the prior art. Specifically, Patent Owner asserts the prior art did not "employ, in a method for providing updates as part of a file backup, a file segmenting method in which segments of a file

[10] The argument in Patent Owner's MTA under the section heading "Patentability of Proposed Claim 47" is substantively identical to the portions of Dr. Mohaptra's Declaration (Ex. 2009) cited in that section. *See* Ex. 2009 ¶¶ 38–41.

are defined by segment delimiters that are statistically determined to be optimal division points for the segments." *Id.* at 14 (citing Ex. 2009 ¶ 41).

Petitioners argue Patent Owner has not met its burden of establishing that it is entitled to the relief requested in the MTA because (1) Patent Owner has not shown the added limitation is patentable over the prior art of record in this proceeding, (2) Patent Owner has not shown patentability of proposed claim 47 over the prior art in general, because Patent Owner has not pointed to any art other than the references already asserted in this proceeding, and (3) the added limitation in proposed claim 47 is not enabled because the Specification of the '799 Patent fails to provide any explanation, beyond the language recited in proposed claim 47, of how the function is achieved. Opp. MTA 1–3.

We are unpersuaded by Patent Owner's arguments because Patent Owner addresses neither the level of ordinary skill in the art nor the prior art known to Patent Owner generally, with respect to a particular feature it added to original patent claims to form the proposed substitute claim. During oral argument, counsel for Patent Owner stated that he did not "know what somebody [of ordinary skill in the art] would consider or not, but it seems at least in [Petitioners' Declarant] Dr. Hutchinson's opinion," what was known in the file back-up field "would be the relevant inquiry." Hr'g Tr. 37:1–6. Patent Owner's discussion of the prior art with respect to the patentability of the added limitation was restricted to the assertions that the recited segmentation methodology was not done previously "when providing updates as part of a file backup," and that "[n]one of the *cited* references . . . teaches or suggests" the amended limitation. Mot. to Amend 14 (emphasis added). During oral argument, upon inquiry from the Board, counsel for Patent Owner acknowledged that its assertions of patentability were limited to the field of file back-up. Hr'g Tr. 36:19–38:12.

Although Patent Owner is not expected to know of all pre-existing prior art, it is expected to indicate what it does know, particularly with respect to the feature it has proposed to add to the original patent claims, i.e., defining segments "by segment delimiters that are statistically determined to be optimal division points." Instead, Patent Owner relied upon art asserted by Petitioner, which relates to updates for file back-up. Petitioner cannot be expected to select the closest art for the proposed additional limitation when Petitioner could not have predicted the limitation(s) Patent Owner would opt to add to proposed claim amendments. The assertion that the added limitation was not taught was restricted to the field of file back-up in Patent Owner's Motion to Amend and Dr. Mohaptra's Declaration. At oral argument, Patent Owner asserted for the first time that it was unaware of *any* art that taught the proposed additional segmentation methodology limitation. *See* Hr'g Tr. 36:9–11, 37:9–11. The statement by counsel for Patent Owner at oral argument, however, is too late to be considered because no new argument may be presented at oral hearing. Patent Owner's Declarant, Dr. Mohaptra, limited his statement that the proposed limitation had not been used to the field of file back-up and nothing in Patent Owner's submitted papers or evidence expanded that assertion to include all relevant prior art. *Id.* at 38:1–12.

For the foregoing reasons, Patent Owner has not met its burden of showing the patentability of the proposed substitute claim 47 over the prior art. Due to the deficiencies in Patent Owner's Motion, we need not and do not consider the Opposition.

Accordingly, the contingent Motion to Amend is *denied*.

III. CONCLUSION

Petitioners have demonstrated by a preponderance of the evidence that: claims 1, 12, 23, 24, 30, 31, 37, and 42 are anticipated by Williams; claims 5–10 and 16–21 are unpatentable as obvious over Williams and Miller; claims 37 and 42 are anticipated by Balcha; claims 1, 5, 9, 10, 12, 16, 20, 21, 23, 24, 30, and 31 are unpatentable as obvious over Balcha and Miller; claims 6–8 and 17–19 are unpatentable as obvious over Balcha, Miller, and Freivald; and claims 37 and 42 are unpatentable as obvious over Balcha and Freivald.

IV. ORDER

In consideration of the foregoing, it is:

ORDERED that claims 1, 5–10, 12, 16–21, 23, 24, 30, 31, 37, and 42 of the '799 patent are held *unpatentable*;

FURTHER ORDERED that Patent Owner's Motion to Amend is *denied*, and

FURTHER ORDERED that, because this is a final written decision, parties to the proceeding seeking judicial review of the decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

For Petitioner:

Michael Kiklis
Scott McKeown
Christohper Ricciuti
cpdocketkiklis@oblon.com
cpdocketmckeown@oblon.com
cpdocketricciuti@oblon.com

For Patent Owner:

Tarek Fahmi
Amy Embert
tarek.fahmi@ascendalaw.com
amy.embert@ascendalaw.com