United States Court of Appeals for the Federal Circuit

PERSONAL WEB TECHNOLOGIES, LLC,
Appellant

v.

APPLE, INC.,
Appellee

2018-1599

Appeal from the United States Patent and Trademark Office, Patent Trial and Appeal Board in No. IPR2013-00596.

Decided: March 8, 2019

LAWRENCE MILTON HADLEY, Glaser Weil Fink Howard Avchen & Shapiro LLP, Los Angeles, CA, argued for appellant. Also represented by JOEL LANCE THOLLANDER, McKool Smith, PC, Austin, TX.

MICHAEL JAY, DLA Piper LLP (US), Los Angeles, CA, argued for appellee. Also represented by NANDAN PADMANABHAN.

Before MOORE, TARANTO, and CHEN, *Circuit Judges.*

CHEN, *Circuit Judge.*

PersonalWeb Technologies, LLC (PersonalWeb) appeals the final written decision of the Patent Trial and Appeal Board (Board) in an *inter partes* review (IPR) proceeding concluding that Apple demonstrated by a preponderance of the evidence that claims 24, 32, 81, 82, and 86 of U.S. Patent No. 7,802,310 (the '310 patent) are unpatentable as obvious in view of two prior art references—U.S. Patent No. 5,649,196 (Woodhill) and U.S. Patent No. 7,359,881 (Stefik). Because one of the Board's key underlying fact findings as to Woodhill's disclosure is not supported by substantial evidence, we reverse.

BACKGROUND

A. The '310 Patent

The '310 patent explains that in conventional data processing systems, data items such as files are typically identified by their user-created alphanumeric name and/or pathname or location. J.A. 69 at 1:53–2:5. Certain problems arise, however, using traditional naming conventions. For example, if one device transfers a data item to a second device using just the name associated with the data item, it is possible that the data item already exists on the second device, and a duplicate of the data item will be created. J.A. 69–70 at 2:63–3:9. The '310 patent contemplates a method and apparatus for resolving this and other concerns by creating a substantially unique identifier for each data item in the data processing system that is independent of the data item's user-defined name, location, etc., but rather is dependent on only the content of the data item itself. J.A. 70 at 3:52–58. The identifier for a particular data item is created by applying a cryptographic hash function to the data item. J.A. 74 at 12:21–26. The output of the hash function is the content-based identifier or "True Name," which is "virtually guaranteed" to be unique to the data item. *Id.* The system uses the content-based identifier alone to determine whether a particular data item is

present on the system. J.A. 70 at 3:59–62. When the data item's contents are changed, the content-based identifier of the data item also changes. *See* J.A. 86 at 35:55–63.

The '310 patent explains that content-based identifiers can be used for various purposes in data processing systems, including, for example, to identify data items in a "license table." J.A. 74 at 11:33–43. The patent describes a license table as a two-field database containing a list of content-based identifiers and, for each content-based identifier, a list of users authorized to access the data item associated with the content-based identifier. *Id.*

Claim 24 is illustrative:

> 24. A computer-implemented method implemented at least in part by hardware comprising one or more processors, the method comprising:
>
> (a) using a processor, receiving at a first computer from a second computer, a request regarding a particular data item, said request including at least a content-dependent name for the particular data item, the content-dependent name being based, at least in part, on at least a function of the data in the particular data item, wherein the data used by the function to determine the content-dependent name comprises at least some of the contents of the particular data item, wherein the function that was used comprises a message digest function or a hash function, and wherein two identical data items will have the same content-dependent name; and
>
> (b) in response to said request:
>
> (i) causing the content-dependent name of the particular data item to be compared to a plurality of values;
>
> (ii) hardware in combination with software determining whether or not access to the

> particular data item is unauthorized based on whether the content-dependent name of the particular data item corresponds to at least one of said plurality of values, and
>
> (iii) based on said determining in step (ii), not allowing the particular data item to be provided to or accessed by the second computer if it is determined that access to the particular data item is not authorized.

J.A. 88.

B. Initial IPR Proceedings

In September 2013, Apple filed a petition requesting an IPR of claims 24, 32, 70[1], 81, 82, and 86 of the '310 patent, asserting multiple grounds of unpatentability. PersonalWeb filed a preliminary patent owner response, and, in March 2014, the Board instituted review on the ground that the petition raised a reasonable likelihood that the challenged claims were unpatentable under 35 U.S.C. § 103(a) for obviousness over Woodhill in view of Stefik. After PersonalWeb filed a patent owner response and Apple filed a reply, the Board held a hearing. In March 2015, the Board issued a final written decision concluding that Apple had demonstrated by a preponderance of the evidence that the challenged claims were unpatentable under § 103(a) over Woodhill in view of Stefik.

C. Woodhill

Woodhill discloses a distributed management system for backing up and restoring data files. *See* J.A. 1674 at 1:11–17. In Woodhill, files are apportioned into 1 MB[2]

---

[1] Though part of the IPR process, claim 70 is not before us on appeal.

[2] The final binary object in the file may be less than 1 MB in size.

"binary objects," which in some instances are apportioned even further into 1 KB[3] "granules." J.A. 1675 at 4:21–30, J.A. 1677 at 7:47–59, J.A. 1680–81 at 14:65–15:4. Woodhill explains that the system uses "Binary Object Identifiers," or for granules, "contents identifiers," to determine whether a binary object or granule has changed from one version of the file to the next. J.A. 1678 at 9:9–27, J.A. 1682 at 17:50–64. Only those binary objects or granules whose content has changed need to be backed up, thereby reducing the amount of data that needs to be transmitted during a backup procedure. *Id.* at 9:6–9, 9:23–27, J.A. 1680–81 at 14:53–15:8. Woodhill explains that every new or changed binary object is backed up onto a remote backup file server, and that a compressed copy of every binary object that the system would need in order to restore a current version of a file to a previous version of the file is stored somewhere on the local area network other than on the local computer. J.A. 1678 at 9:30–44.

Woodhill discloses a File Database containing three levels of records. J.A. 1675 at 3:45–54. At the highest level, File Identification Records are stored for each file that has been backed up by the system. *Id.* at 3:54–56. Each File Identification Record includes, *inter alia*, the file's name and location. *Id.* at 3:57–63. At the second level, each File Identification Record is associated with one or more Backup Instance Records, each of which contains information about a backup version of the file. *Id.* at 3:64–4:2. Each Backup Instance Record includes, *inter alia*, a link to the File Identification Record. *Id.* at 4:2–11. At the third level, each Backup Instance Record is associated with one or more Binary Object Identification Records (depending on the size of the file). *Id.* at 4:12–47. A Binary Object Identification Record is created for each binary object of the

---

[3] The final granule in the binary object may be less than 1 KB in size.

backup file version. *Id.* at 4:30–34. Each Binary Object Identification Record includes, *inter alia*, a link to the Backup Instance Record, the Binary Object Size, the Binary Object Offset, and the Binary Object Hash. *Id.* at 4:35–43.

The Binary Object Size and the Binary Object Hash, together with other information not relevant to this appeal, make up the "Binary Object Identifier" for that version of the binary object. *Id.* at 4:43–47. The Binary Object Identifier is unique to a particular binary object because of the Binary Object Hash field, which is created by applying a cryptographic hash algorithm to contents of the binary object. J.A. 1677 at 8:22–35. Woodhill explains that the critical feature of the Binary Object Identifier is that, because it is based on the contents of the binary object, the identifier changes when the contents of the binary object change. *Id.* at 8:58–62. Duplicate binary objects can therefore be recognized from their identical Binary Object Identifiers even if they reside on different types of computers in the same network. *Id.* at 8:62–65.

D. Stefik

Stefik discloses an authentication system for controlling access to digital works. J.A. 1632 at 3:58–4:12. Each digital work's unique identifier and associated usage rights, among other information, are stored in a repository. J.A. 1635 at 9:15–61. A user accesses a digital work over a network using a "digital ticket," which entitles the ticket holder to exercise usage rights associated with the work because, for example, the user has paid for access. J.A. 1632 at 3:59–64.

E. First Federal Circuit Appeal

PersonalWeb appealed the 2015 Board unpatentability decision to our court, and on February 14, 2017, we issued an opinion affirming the Board's claim construction and vacating and remanding the Board's obviousness finding for

further consideration. *See Pers. Web Techs., LLC v. Apple, Inc.*, 848 F.3d 987 (Fed. Cir. 2017) (*PersonalWeb I*). We noted that none of the parties disagreed that Woodhill's Binary Object Identifier corresponded to the claimed content-based identifier[4] for a data item of the '310 patent. *Id.* at 991. However, we determined that the Board's analysis with respect to obviousness—including both whether Woodhill and Stefik disclosed all of the elements recited in the challenged claims and whether a skilled artisan would have been motivated to combine them in the manner recited in the '310 patent—was inadequate. *Id.* at 993.

The main claim element in dispute in that appeal, as in this appeal, was claim 24's "causing the content-dependent name of the particular data item to be compared to a plurality of values." *Id.* The Board had cited only Stefik as satisfying this element, but Apple made clear in its petition that it relied on only Woodhill for this element. *Id.* We disagreed with the Board's use of Stefik and instructed the Board to evaluate whether column 17 of Woodhill, which was the only portion of Woodhill cited by Apple in its petition, taught this element. *Id.* We also disagreed with the Board's motivation-to-combine analysis, which merely affirmed Apple's allegation that a skilled artisan "would have allowed for the selective access features of Stefik to be used with Woodhill's content-dependent identifiers feature." *Id.* (emphasis omitted). We explained that this reasoning said nothing more than that the two references *could* be combined, not that there would been a motivation to combine them, and lacked any explanation as to *how* the

---

[4] In *PersonalWeb I*, we used the term "content-based identifier" to refer to multiple claim terms including "content-dependent name," "content-based identifier," and "digital identifier," because no issue before us turned on any differences between them. *Id.* at 990. We do the same here.

combination of the two references was supposed to work. *Id.* at 993–94. We remanded for the Board to reconsider the merits of the obviousness challenge. *Id.* at 994.

The Board ordered additional briefing by the parties to explain where Apple did or did not make a proper case of obviousness on the instituted ground. J.A. 4.

F. Board's Remand Analysis

On remand, the Board maintained the same obviousness theory of unpatentability, except that the Board replaced its previous reliance on Stefik for teaching the "compared to a plurality of values" element with reliance on column 17 of Woodhill. J.A. 13–16. The Board also expanded its analysis of a skilled artisan's motivation to combine Woodhill and Stefik. J.A. 16–18.

Element (b)(i) of claim 24 is the main point of contention between the parties. It recites, after "in response to said request," "causing the content-dependent name of the particular data item to be compared to a plurality of values." J.A. 88 at 40:15–17. The Board pointed to Apple's petition, which cited column 17, lines 40 to 46 of Woodhill:

> Program control then continues with step 446 where the Distributed Storage Manager program 448 transmits an "update request" to the remote backup file server 12 which includes the Binary Object Identification Record 58 for the previous version of each binary object as well as the list of "contents identifiers" calculated in step 444.

J.A. 14. The Board continued to point to Apple's petition, which cited Apple's expert, and stated: "in order to determine which data needs to be restored by the update request, the remote backup file server of Woodhill *must* be able to reference its local files using the information it receives - namely the Binary Object Identification Record." *Id.* (emphasis added) (brackets and internal quotation marks omitted). The Board then adopted Apple's expert's

conclusion that the "referencing *necessarily must* be accomplished" using "the remote backup fileserver," which "maintains some sort of file system or other mapping (i.e., a database) that allows the Binary Object Identification Record to serve as a lookup for the requisite file data that is to be restored." *Id.* (emphasis added). The Board therefore concluded that Woodhill, without saying so, necessarily compares the content-based identifier (Binary Object Identifier) of the particular data item (binary object) to a plurality of values (an unmentioned but necessarily present database of Binary Object Identifiers). In other words, the Board found that Woodhill inherently teaches comparing a Binary Object Identifier to a plurality of Binary Object Identifiers.

As to Stefik, the Board described Stefik as "a system that addresses the problem of preventing unauthorized access to digital works with an access request utilizing a unique identifier for the digital work." J.A. 15. The Board stated that it "agree[d] with Apple that access provided in Stefik would necessarily require a comparison between the unique identifier and other values to see if a match can be obtained," citing Apple's expert testimony as support. J.A. 15–16. Apple further contended that "a skilled artisan would have combined the backup and restore system in Woodhill with the repository in Stefik to add an authorization layer to prevent unauthorized users from accessing a different user's back up files." J.A. 16. The Board concluded that this rationale was sufficient, and that Apple had demonstrated by a preponderance of the evidence that the challenged claims were unpatentable under § 103(a) as obvious over Woodhill and Stefik. J.A. 17–18.

PersonalWeb appealed both the Board's inherency finding and its motivation-to-combine finding. We have jurisdiction under 28 U.S.C. § 1295(a)(4)(A).

DISCUSSION

"We review the Board's ultimate determination of obviousness de novo and its underlying factual determinations for substantial evidence." *PersonalWeb I*, 848 F.3d at 991. "On the factual components of the inquiry, we ask whether a reasonable fact finder could have arrived at the agency's decision, which requires examination of the record as a whole, taking into account evidence that both justifies and detracts from an agency's decision." *Id.* (internal brackets and quotation marks omitted).

We conclude that the Board's inherency finding derived from column 17 of Woodhill for teaching the "compared to a plurality of values" limitation lacks substantial evidence. While it is possible that Woodhill's system utilizes an unstated Binary Object Identifier lookup table to locate binary objects of a previous version of a file that is going to be restored (column 17 of Woodhill), mere possibility is not enough. "Inherency . . . may not be established by probabilities or possibilities." *PAR Pharm., Inc. v. TWI Pharm., Inc.*, 773 F.3d 1186, 1195 (Fed. Cir. 2014). "The mere fact that a certain thing *may* result from a given set of circumstances is not sufficient." *Id.* (emphasis added). Rather, a party must "show that *the natural result flowing* from the operation as taught would result in the performance of the questioned function." *Id.* (emphasis in original).

As PersonalWeb suggests, an equally plausible, if not more plausible, understanding of Woodhill is that Woodhill's system uses conventional file names and locations to locate files and the Binary Object Offset field to locate a given binary object within a file. Before the passage relied on by Apple and the Board (lines 40 to 46), column 17 states that Woodhill's system "obtains *from the user* the identities of the current and previous versions of the file (comprised of binary objects) which needs to be restored." J.A. 1682 at 17:28–32 (emphasis added); *see also* J.A. 1670 at Fig. 5I (step 442). The next sentence confirms

that the file is "user-specified." J.A. 1682 at 17:32–35. The Board's proffered look-up table is therefore unnecessary to locate the current or previous version of the file.

Even if the file was not specified by the user, Woodhill's only disclosed method of locating a current or previous file is by searching for the file using standard file block information, including the file name and location. J.A. 1676 at 5:46–6:11. Woodhill does not disclose searching for a file based on a content-based identifier.

As to locating binary objects within a file, Woodhill explains that the system compiles a list of these binary objects using "information . . . obtained from File Database 25" after the identity of the file is obtained. J.A. 1682 at 17:32–36, J.A. 1670 at Fig. 5I (step 443). Although this portion of Woodhill's specification does not specify what exactly in the File Database is used to locate a particular binary object within a given file, Woodhill explains in column 9 that binary objects can be located within a particular file using the Binary Object Stream Type field and the Binary Object Offset field of each Binary Object Identification Record. J.A. 1678 at 9:14–22. That disclosure, albeit at a different location in Woodhill's written description than column 17, suggests that Woodhill contemplated a means for locating binary objects that would make Apple and the Board's proposed look-up table unnecessary for Woodhill's restore process described at column 17. As PersonalWeb correctly points out, the only disclosed use of Woodhill's Binary Object Identifier is to perform a one-to-one comparison with the Binary Object Identifier associated with the backed-up version of the binary object, which occurs after the appropriate binary object has been located, according

to column 9. Neither Apple nor its expert provided any adequate response to this reading of Woodhill.[5]

Because we find that the proposed, theoretical Binary Object Identifier look-up table that Apple and the Board rely on does not necessarily exist in Woodhill, the Board's reliance on inherency for that element in its obviousness analysis was improper. Apple provided no other basis for this element of comparing a content-based identifier to a plurality of values being disclosed or otherwise obvious to a skilled artisan at the time of the invention. We therefore reverse the Board's finding of obviousness over Woodhill in view of Stefik. We need not reach the question of motivation to combine. We have considered the parties' remaining arguments and find them unpersuasive.

REVERSED

---

[5] We also note that the binary object look-up table proffered by Apple and the Board could lead to a peculiar design if it contained multiple binary objects for every large file on the system, as Apple's counsel alleged at oral argument, thereby requiring a comparison of a particular Binary Object Identifier to millions of Binary Object Identifiers not even associated with the correct file. *See* Oral Argument at 14:20–15:50 (acknowledging that there could be ten million Binary Object Identifiers in the look-up table for a far smaller number of files).