

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

DROPBOX, INC.,
Petitioner,

v.

ENTANGLED MEDIA, LLC,
Patent Owner.

IPR2024-00285
Patent 8,484,260 B2

Before MICHELLE N. WORMMEESTER, JOHN A. HUDALLA, and
DONALD E. DAYBELL, *Administrative Patent Judges*.

DAYBELL, *Administrative Patent Judge*.

DECISION
Denying Institution of *Inter Partes* Review
35 U.S.C. § 314

I. INTRODUCTION

Dropbox, Inc. (“Petitioner”) filed a Petition requesting *inter partes* review of claims 1–8 (“challenged claims”) of U.S. Patent No. 8,484,260 B2 (Ex. 1001, “the ’260 Patent”) pursuant to 35 U.S.C. §§ 311–319. Paper 2 (“Pet.”). Entangled Media, LLC (“Patent Owner”) filed a Preliminary Response. Paper 6 (“Prelim. Resp.”).

We have authority under 35 U.S.C. § 314 and 37 C.F.R. § 42.4 to determine whether to institute review. Upon consideration of the arguments and evidence before us, we conclude that Petitioner has not shown a reasonable likelihood that it would prevail in establishing unpatentability of any challenged claim of the '260 Patent.

The Petition is *denied*, and no *inter partes* review is instituted.

A. Real Parties in Interest

Petitioner identifies itself (Dropbox, Inc.) as real party in interest. Pet. 1. Patent Owner identifies Entangled Media, LLC and Entangled Media Corporation as real parties in interest. Paper 5, 1 (Mandatory Notices).

B. Related Matters

Each party identifies the following related matters:

Entangled Media, LLC v. Dropbox, Inc., No. 5:23-cv-03264-PCP (N.D. Cal.) (“the underlying litigation”) and *Ex Parte* Reexamination Proceeding, Control No. 90/015,221. Pet. 1; Paper 5, 1 (Mandatory Notices).¹ Petitioner additionally identifies IPR2024-00284 (challenging related U.S. Patent No. 8,296,338). Pet. 1.

*C. The '260 Patent*²

The '260 Patent addresses a problem that arises when a user stores files on multiple different devices. Ex. 1001, 1:22–29. In order for a user to

¹ The '221 Reexamination was concluded on January 30, 2024, by issuance of a Reexamination Certificate confirming the patentability of all claims challenged therein. Ex. 3001.

² The '260 Patent issued from application No. 13/424,366, filed March 19, 2012, which is a divisional of Application No. 12/774,231, filed May 5, 2010, now Pat. No. 8,296,338. Ex. 1001, codes (21), (22), (62). The '260 Patent also claims the benefit of Provisional Patent Application No. 61/175,489, filed May 5, 2009. *Id.* at code (60).

be able to access files stored on different devices, it is desirable to synchronize those files in some manner. *Id.* at 2:7–15. However, synchronizing files across multiple devices creates an issue with consumption of storage space. Ex. 1001, 1:54–2:2. Making copies of a user’s files on each device consumes a large amount of storage space. *Id.* The invention disclosed in the ’260 Patent seeks to make all of a user’s files available on any of the user’s devices, while reducing the amount of storage space consumed by those files. *Id.* at 1:55–2:15. To accomplish this, the ’260 Patent discloses a system and process that creates a “singular file system,” which is a unified representation of all the user’s data across all of the user’s devices. *Id.* at 2:16–20, 6:8–21. This singular file system aggregates all of the user’s files into a single hierarchy, regardless of which device each file is stored on. *Id.* at 9:15–22, Fig. 6.

Figure 6 of the ’260 Patent is reproduced below.

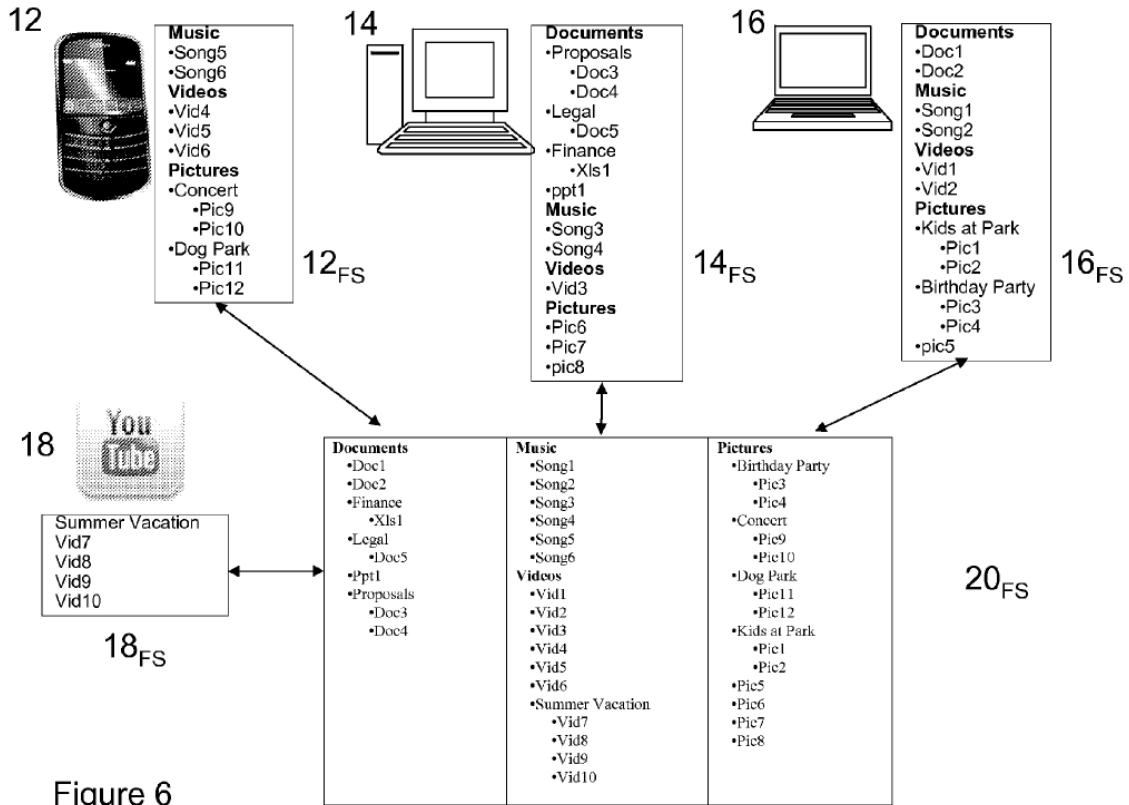


Figure 6

Figure 6 depicts exemplary unified file structure 20_{UFS}.³ Ex. 1001, 4:8–9, 9:15–16. Unified file structure 20_{UFS} incorporates data from user devices, including PDA 12, laptop computer 14, and desktop computer 16, and from online service 18. *Id.* at 4:59–5:10, 9:15–22. In particular, unified file structure 20_{UFS} is created from individual device file structures 12_{UFS}, 14_{UFS}, and 16_{UFS}, and online service file structure 18_{UFS}. *Id.* at 9:15–19. Unified file structure 20_{UFS} then replaces device file structures 12_{UFS}, 14_{UFS}, and 16_{UFS} in devices 12, 14, and 16. *Id.* at 9:19–21. In contrast, the file structure of online service 18 is not unified. *Id.* at 9:21–22.

In the singular file system of the '260 Patent, all of the user's files are displayed, regardless of which device those files are stored on, and

³ The reference numerals in Figure 6 use the subscript “FS” while the reference numerals in the written description use the subscript “UFS.”

regardless of which device is used to view and access the files. *Id.* at 6:19–31 (“The metadata instructs the software client where to store the data files, ensuring that a common view and storage location is maintained across all devices.”). To minimize the amount of space required to store all of a user’s files, the file system does not duplicate the user’s files. *Id.* at 2:22–24. Instead, the file system stores each file just once, on one of the user’s devices. *Id.* at 2:24–30. The rest of the user’s devices store only a virtual copy of the files that are physically stored on another device. *Id.* Then, with the unified file structure depicting all of the user’s files and implemented on each of the user’s devices, the user can simply request a desired file. *Id.* at 8:30–35. The user’s device receives this request, and software on the device intercepts the received request. *Id.* at 8:35–45. This software determines whether the file is stored physically on the device, or is a virtual file that is physically stored elsewhere. *Id.* If the file is stored elsewhere, then a web service creates a peer-to-peer connection from the requesting device to the device containing the physical file. *Id.* at 8:45–52. The file is copied across this connection and stored on the requesting device, where it can be further operated on. *Id.* at 8:54–60.

D. Illustrative Claims

Claims 1, 7, and 8 are independent and reproduced below.⁴

1[P]: A process for operating on files located on multiple devices using a singular file system comprising:

⁴ The heading with bracketed letters identifying each of the claim elements correspond to those used by Petitioner to reference the claim elements. *See* Pet. App. A. We use them here for ease of reference, understanding, and consistency.

- 1[A]: accepting a request to operate on a file at a first device, wherein the file is selected from the singular file system on the first device;
- 1[B]: modifying the singular file system on the first device to make local files and virtual files appear indistinguishable to the singular file system, the local files and virtual files sharing a same location on the first device;
- 1[C]: intercepting the request by a software client on the first device;
- 1[D]: determining by the software client if the file is physically located on the first device or if the file is a virtual file of a corresponding file physically stored on a second device by reviewing file metadata,
- 1[E]: wherein a visual representation of the singular file system on the first device is identical to a visual representation of the singular file system on the second device; and
- 1[F]: if the file is the virtual file of the corresponding file physically located on the second device, requesting by the software client on the first device that a peer-to-peer connection be brokered by a server-based web service between the first device and the second device;
- 1[G]: if the peer-to-peer connection is brokered, transferring the corresponding physical file from the second device to the first device; and
- 1[H]: performing the operation on the transferred corresponding physical file at the first device.

Ex. 1001, 11:25–52.

- 7[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:
- 7[A]: accepting a request to operate on a file at a first device, wherein the file is selected from the singular file system on the first device;

- 7[B]: modifying the singular file system on the first device to make local files and virtual files appear indistinguishable to the singular file system, the local files and virtual files sharing a same location on the first device;
- 7[C]: intercepting the request by a software client on the first device;
- 7[D]: determining by the software client if the file is physically located on the first device or if the file is a virtual file of a corresponding file physically stored on a second device by reviewing file metadata,
- 7[E]: wherein a visual representation of the singular file system on the first device is identical to a visual representation of the singular file system on the second device; and
- 7[F]: if the file is the virtual file of the corresponding file physically located on the second device, requesting by the software client on the first device that a peer-to-peer connection be brokered by a server-based web service between the first device and the second device;
- 7[G]: if the peer-to-peer connection is brokered, transferring the corresponding physical file from the second device to the first device; and
- 7[H]: performing the operation on the transferred corresponding physical file at the first device.

Id. at 12:5–35.

- 8[P]: A client comprising:
- 8[B]⁵: a memory;
- 8[C]: at least one processor configured to:

⁵ Petitioner has skipped over 8[A]. Pet. App A. To avoid confusion, we adopt Petitioner’s references for this claim.

- 8[D]: accept a request to operate on a file at a first device, wherein the file is selected from the singular file system on the first device;
- 8[E]: modify the singular file system on the first device to make local files and virtual files appear indistinguishable to the singular file system, the local files and virtual files sharing a same location on the first device;
- 8[F]: intercept the request by a software client on the first device;
- 8[G]: determine by the software client if the file is physically located on the first device or if the file is a virtual file of a corresponding file physically stored on a second device by reviewing file metadata,
- 8[H]: wherein a visual representation of the singular file system on the first device is identical to a visual representation of the singular file system on the second device; and
- 8[I]: if the file is the virtual file of the corresponding file physically located on the second device, request by the software client on the first device that a peer-to-peer connection be brokered by a server-based web service between the first device and the second device;
- 8[J]: if the peer-to-peer connection is brokered, transfer the corresponding physical file from the second device to the first device; and
- 8[J]: perform the operation on the transferred corresponding physical file at the first device.

Id. at 12:36–65.

E. Evidence relied on by Petitioner

Petitioner relies on the following document references.

Name	Reference	Exhibit
Havewala	U.S. Patent Publication 2007/0016621	1005

Adams	U.S. Patent Publication 2002/0046232	1006
Saridakis	U.S. Patent 8,874,691 B2	1007
Rothman	U.S. Patent Publication 2005/0289218	1008

Petitioner also relies on the Declaration of Dr. Patrick McDaniel (Ex. 1003).

F. Prior Art and Asserted Grounds

Petitioner asserts that claims 1–8 are unpatentable on the following grounds:

Ground	Claim(s) Challenged	35 U.S.C. §⁶	Reference(s)/Basis
1	1, 4–8	103	Havewala, Adams
2	2, 3	103	Havewala, Adams Saridakis
3	1–8	103	Havewala, Adams, Saridakis, Rothman

II. ANALYSIS

A. Level of Ordinary Skill in the Art

Petitioner asserts that a person of ordinary skill in the art (“POSITA”) as of the time of the invention of the ’260 Patent “would have been a person holding at least a master’s degree in computer science or electrical engineering, or a related degree, and with at least two years [of] training or experience with networking and file systems.” Pet. 10. Petitioner’s declarant, however, indicates that a bachelor’s degree is sufficient to qualify as a POSITA. Ex. 1003 ¶ 21.

⁶ The Leahy-Smith America Invents Act (“AIA”), Pub. L. No. 112-29, 125 Stat. 284, 287–88 (2011), amended 35 U.S.C. §§ 102 and 103. Because the ’260 patent was filed before March 16, 2013 (the effective date of the relevant amendments), the pre-AIA versions of §§ 102 and 103 apply.

Patent Owner objects to Petitioner's requirement that a POSITA must have a master's degree. Prelim. Resp. 3. Patent Owner points out the discrepancy between Petitioner and its own declarant, as well as with the position Petitioner allegedly took in the underlying litigation. *Id.* (citing Ex. 1003 ¶ 21; Ex. 2001, 14–15).

At this juncture, the parties only appear to dispute the educational background in the definition of the level of ordinary skill in the art. The only evidence in the present record regarding the educational background consists of Dr. McDaniel's declarations in this case and in the underlying litigation. *See* Ex. 1003 ¶ 21; Ex. 2001, 14–15. We agree with Patent Owner that both of these declarations support that a person of ordinary skill in the art would have had either a bachelor's or a master's degree in computer science or electrical engineering or a related degree. Prelim. Resp. 2–3. Moreover, Petitioner does not explain why a master's degree is necessary. Thus, we revise Petitioner's proposed definition to conform to Dr. McDaniel's testimony. We additionally remove the qualifier "at least" from Petitioner's proposed definition, because this qualifier introduces ambiguity.

Thus, for purposes of this Decision, we apply the following definition of the level of ordinary skill in the art: a POSITA would have been a person holding a bachelor's degree or a master's degree in computer science or electrical engineering or a related degree, with two years of training or experience with networking and file systems. This definition appears to be consistent with what is reflected by the content of the applied prior art references as well. *Cf. Okajima v. Bourdeau*, 261 F.3d 1350, 1354–55 (Fed. Cir. 2001) (the applied prior art may reflect an appropriate level of skill). We note, however, that our analysis below would not change even if we adopted Petitioner's definition as stated in the Petition.

B. Claim Construction

We use “the same claim construction standard that would be used to construe the claim in a civil action under 35 U.S.C. [§] 282(b), including construing the claim in accordance with the ordinary and customary meaning of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent.” 37 C.F.R. § 42.100(b) (2023). The claim construction standard set forth in *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc) is applicable.

Claim terms are generally given their ordinary and customary meaning as would be understood by one with ordinary skill in the art in the context of the specification, the prosecution history, other claims, and extrinsic evidence including expert and inventor testimony, dictionaries, and learned treatises, although extrinsic evidence is less significant than the intrinsic record. *Phillips*, 415 F.3d at 1312–17. Usually, the specification is dispositive, and “it is the single best guide to the meaning of a disputed term.” *Id.* at 1315.

The specification may reveal a special definition given to a claim term by the patentee, or the specification or prosecution history may reveal an intentional disclaimer or disavowal of claim scope by the inventor. *Id.* at 1316. If an inventor acts as his or her own lexicographer, the definition must be set forth in the specification with reasonable clarity, deliberateness, and precision. *Renishaw PLC v. Marposs Societa' per Azioni*, 158 F.3d 1243, 1249 (Fed. Cir. 1998). A disavowal of claim scope, if any, can be effectuated by language in the specification or the prosecution history. *Poly-America, L.P. v. API Indus., Inc.*, 839 F.3d 1131, 1136 (Fed. Cir. 2016).

Only those claim terms that are in controversy need to be construed, and only to the extent necessary to resolve the controversy. *Nidec Motor*

Corp. v. Zhongshan Broad Ocean Motor Co., 868 F.3d 1013, 1017 (Fed. Cir. 2017); *Wellman, Inc. v. Eastman Chem. Co.*, 642 F.3d 1355, 1361 (Fed. Cir. 2011); *Vivid Techs., Inc. v. Am. Sci. & Eng'g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999).

Petitioner asserts that it “does not propose that any terms require construction.” Pet. 11. Similarly, Patent Owner asserts that it “does not propose that the Board expressly construe any claims.” Prelim. Resp. 4. For purposes of this Decision, we determine that no claim term requires express interpretation to resolve any controversy in this proceeding.

C. *Havewala in view of Adams, Claims 1 and 4–8 (Ground 1)*

1. *The Law on Obviousness*

A claim is unpatentable under 35 U.S.C. § 103 if the differences between the claimed subject matter and the prior art are such that the subject matter, as a whole, would have been obvious to a person having ordinary skill in the art to which said subject matter pertains. *See KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). The question of obviousness is resolved on the basis of underlying factual determinations, including (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of skill in the art; and (4) where in evidence, so-called secondary considerations, including commercial success, long-felt but unsolved needs, failure of others, and unexpected results.⁷ *Graham v. John Deere Co. of Kansas City*, 383 U.S. 1, 17–18 (1966).

⁷ The parties do not rely on any evidence of secondary considerations. Accordingly, we do not consider this factor in our analysis.

2. *Havewala*

Havewala is a U.S. patent application publication directed to an architecture and method that allow data from a computer file stored on a storage volume to be moved or “ghosted” to an alternate location, in order to free space on the storage volume. Ex. 1005 ¶ 8. Havewala addresses a problem where certain of a user’s files can become of less interest to the user. *Id.* ¶ 10. In such circumstances, the user’s files are considered to be “cold,” and it is undesirable to continue to store them on the user’s device. *Id.* Havewala teaches that in such circumstances, the user’s cold files can be moved off of the user’s device and onto an alternate location. *Id.* ¶ 11. Havewala further teaches that when a user’s file is moved, a “ghost” of that file is left behind on the user’s device. *Id.* ¶ 50. This “ghosted” file is a stub that has some file metadata, but no actual data in it. *Id.*

Figure 2 of Havewala is reproduced below.

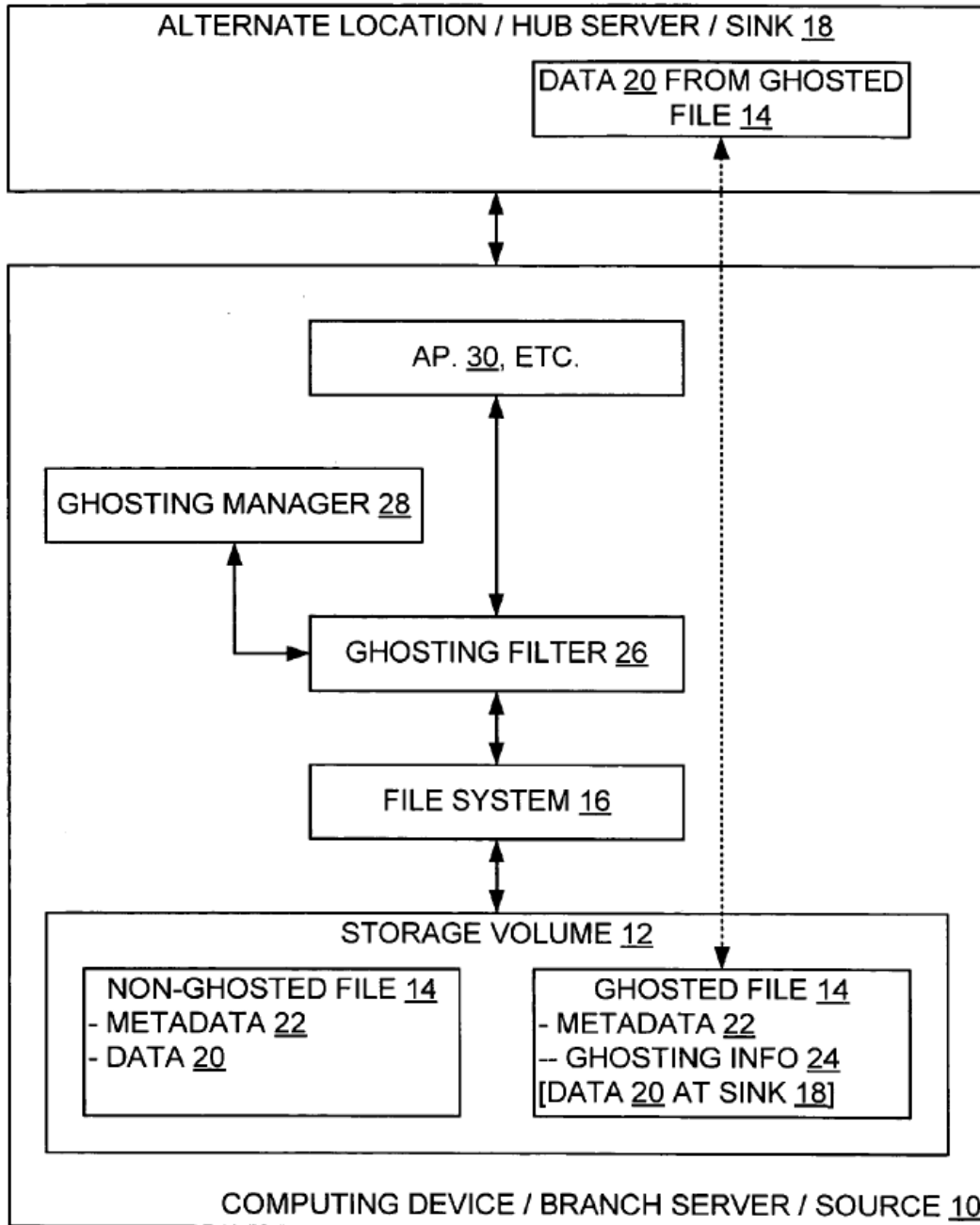


Fig. 2

Figure 2 depicts source 10 (corresponding to a user's device) storing both ghosted files 14 and non-ghosted files 14 in storage volume 12. Ex. 1005 ¶ 48. These files are accessed by the device's file system 16, which in turn interfaces with ghosting filter 26 and ghosting manager 28. *Id.* ¶¶ 64–65.

Ghosted files 14 have their data 20 removed from them and stored at sink 18 (which can be an alternate location such as a server). *Id.* ¶ 48. Non-ghosted files 14 remain on device 10, and are not copied back to sink 18. *Id.* ¶ 52.

When a user wishes to access a ghosted file, the user makes a request to “open” the file. Ex. 1005 ¶¶ 64, 67. This request is sent to Havewala’s file system. *Id.* ¶ 68. Since the ghosted file has no data, the file system returns an error message that is intercepted by a ghosting filter. *Id.* The ghosting filter perceives that the requested file has been ghosted based on the fact that an error message was generated. *Id.* The ghosting filter then reads the ghosted file’s metadata, determines where the ghosted file is actually stored, and returns a handle to the user that is used for subsequent file accesses. *Id.* When a file is subsequently accessed, for example via a “read” request, Havewala’s system then “reconstitutes” the ghosted file by copying the file’s actual data from the location where it resides back to the user’s device. *Id.* ¶¶ 71–72. Thus, “open” requests are akin to a person identifying and opening up a hardcopy file in order to facilitate access to the contents of the file, and a “read” request is akin to a user picking up some of the contents of that hardcopy file and actually examining them. *Id.* ¶¶ 69, 71.

3. *Adams*

Adams teaches a method of distributed file sharing over a network. Users maintain files on their various devices. Ex. 1006 ¶ 36. Adams’ system categorizes these files into categories, and stores those category lists on a central index server. *Id.* ¶ 15. A user wishing to locate a file will search the central index on the server. *Id.* The server will return a list of matching files, and the locations where those files can be found. *Id.* ¶ 16. A peer-to-peer connection is created to allow the user to access the desired file from another user’s device. *Id.* ¶ 51.

4. *Analysis of Claim 1*

For purposes of this Decision, we need only analyze selected limitations of claim 1, as set forth below.

a) *Limitation 1[A]: accepting a request to operate on a file at a first device*

Petitioner contends that Havewala teaches this limitation. Pet. 20–21. Petitioner contends that Havewala’s system “‘receive[s] [a] request to access’ or open the file that is ghosted on the first device.” *Id.* at 21 (citing Ex. 1005 ¶¶ 59, 60, 66, 67, Fig. 5). Therefore, for this limitation, Petitioner relies on the “open” message sent to the file system as the recited “request.” *Id.* at 20–22 (“Havewala discloses ‘accepting a request to operate on a file at a first device [*e.g.*, Havewala’s open command that is received and accepted by the first computing device or source/branch server 10]’” (alteration by Petitioner)).

As discussed immediately below, however, Petitioner relies on a different feature of Havewala for teaching limitation 1[C], which claims the same “request.”

b) *Limitation 1[C]: intercepting the request by a software client on the first device*

Petitioner contends that Havewala teaches this limitation. Pet. 32. To satisfy the claimed “intercepting the request,” Petitioner chiefly relies on Havewala’s handling of “open” requests. *Id.* at 33–34. Additionally, Petitioner suggests tangentially that Havewala’s “read” requests may also satisfy this limitation. *Id.* at 34. We address each argument in turn.

(1) *Havewala's "open" request*

Petitioner contends that Havewala's handling of an "open" request constitutes "intercepting a request to operate on a file" as claimed. Pet. 33. Specifically, Petitioner points to the following passage of Havewala:

when a user or application 30 tries to access the removed data 20 of a ghosted file 14, the file system 16 upon receiving a **request for such access** will discover that the ghosted file 14 does not contain the removed data 20 and thus *return an error* which the **ghosting filter 26 will intercept**. Upon such **interception**, then, **the ghosting filter 26 employs the file system 16 to obtain the ghosting information 24 from the metadata 22 for the ghosted file 14, and based on the obtained ghosting information 24 such ghosting filter 26 triggers reconstituting of such ghosted file 14** ... so that the request for access may ultimately be honored.

Id. at 33 (italics emphasis added) (quoting Ex. 1005 ¶ 64).

Petitioner thus relies for this limitation on Havewala's interception of an error message generated by the file system, in response to a "request for . . . access," when the file system discovers that the ghosted file 14 does not contain the removed data 20. *Id.* This error message, however, is not the same feature of Havewala that Petitioner relies on as teaching the same claimed "request" in the earlier-recited "accepting a request" limitation.

The instant limitation recites "intercepting the request," which relies for its antecedent basis on the "accepting a request" limitation recited above. Therefore, this claim requires that the same "request" be both accepted and intercepted. *See Baldwin Graphic Sys., Inc. v. Siebert, Inc.*, 512 F.3d 1338, 1342–43 (Fed. Cir. 2008) (noting that definite articles such as "the" and "said" create anaphoric phrases that refer to the initial antecedent phrase, and do not alter the meaning of the antecedent "in the slightest"). Petitioner, however, fails to identify any "request" that is both "accepted" and

“intercepted.” Instead, as discussed above, for the “accepting a request” limitation, Petitioner relies on the “open” message sent to the file system as the recited “request” that is accepted. *Id.* at 20–22. For the “intercepting the request” limitation, however, Petitioner relies on an error message sent back from the file system to the application as the recited “request” that is intercepted. *Id.* at 33. Thus, Petitioner points to two different features of Havewala for teaching the same required “request” in claim 1. This is fatally inconsistent. Therefore, Petitioner fails to adequately demonstrate that Havewala teaches that the same “request” is first accepted and then intercepted, as required by the claim.

Even were we to overlook the inconsistency in Petitioner’s analysis, Petitioner still fails to establish that Havewala teaches the claimed “request to operate on a file.” Petitioner relies for this limitation on an error message generated by the file system as the claimed “request.” Pet. 33. This error message, however, is not a “request to operate on a file,” as claimed. This error message is instead a response generated by the file system to a request from the application or operating system to open a file. Ex. 1005 ¶ 64. The response indicates that the request cannot be fulfilled because the target file has no data. *Id.*

Havewala discusses its messaging paradigm with reference to Figure 6, which is excerpted below:

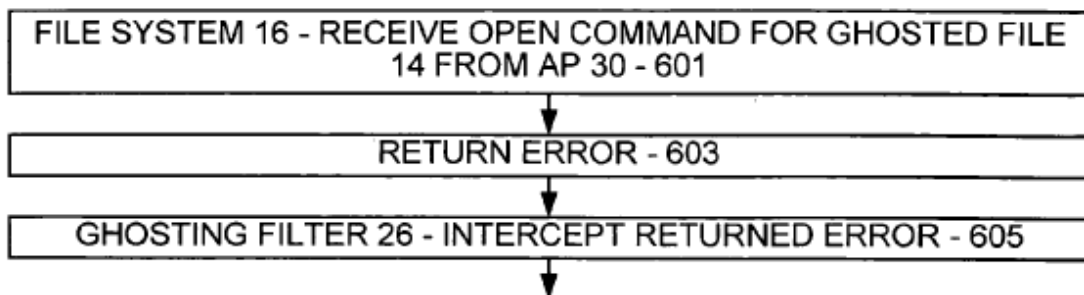


Figure 6 of Havewala, as excerpted, is a flow diagram detailing key steps performed in reconstituting a ghosted file from the sink. *See* Ex. 1005 ¶ 27. This excerpt depicts steps 601, 603, and 605. At step 601, the file system 16 receives an open command from an application 30. *Id.* ¶ 67. In this step, the “file system . . . notes that the ghosted file 14 does not contain the removed data 20.” *Id.* ¶ 68. Then, after the open command has already been received and processed by Havewala’s file system, the file system returns an error at step 603. *Id.* at Fig. 6. This “returned error” is what is intercepted by the ghosting filter 26, at step 605. *Id.* ¶ 68, Fig. 6. Figure 6 of Havewala and the associated text at paragraphs 67 and 68 demonstrate that the error message that is intercepted by Havewala’s ghosting filter is not a request; instead, it is a response to the “open” request. Petitioner’s argument that an error message is the claimed “request” is therefore unsupported by the evidence of record.

(2) *Havewala’s “read” request*

Petitioner additionally suggests that Havewala’s “read” command is relevant to this limitation. Pet. 34 (quoting Ex. 1005 ¶ 71) (“Havewala further discloses that ‘the ghosting filter stores the read command in a queue or the like for awaiting further processing (step 613).’”). To the extent that Petitioner seeks to rely for this limitation on the “read” command, Petitioner again relies on a different feature than it relied on for the same “request” in limitation 1[A]. *See supra*, Section II.C.4.a (relying on Havewala’s “open” command as the claimed request). This inconsistency is fatal to Petitioner’s argument here.

Additionally, Petitioner fails to articulate any further argument about *how* Havewala’s processing of the “read” command allegedly satisfies the “intercepting the request” limitation of claim 1. These failures alone

undermine Petitioner's reliance on Havewala's "read" command. Nevertheless, even were we to find that the "read" command teaches this limitation, this command fails to satisfy the "determining" limitation, as discussed below.

Therefore, Petitioner has not shown sufficiently that Havewala teaches this limitation.

c) Limitation 1[D]: determining by the software client if the file is physically located on the first device or if the file is a virtual file of a corresponding file physically stored on a second device by reviewing file metadata

Petitioner contends that Havewala satisfies this limitation. Pet. 34–39. Patent Owner disputes these contentions. Prelim. Resp. 14–15.

Petitioner contends that Havewala teaches "determining whether a file on the first device is physical or virtual, by a software client using file metadata." Pet. 37–39. According to Petitioner, once a file in Havewala has been ghosted, "the metadata of the now-ghosted file is amended to include ghosting information including information that may be employed to retrieve the moved data for the file from the alternate location." *Id.* at 37 (emphasis omitted) (quoting Ex. 1005 ¶ 19). According to Petitioner, Havewala's ghosting filter then intercepts a request to operate on a ghosted file. *Id.* at 38. Having intercepted the request, Havewala's ghosting filter then uses the file metadata to locate the primary data 20 for the ghosted file 14, as that data is stored at the sink 18. *Id.* This, according to Petitioner, satisfies the "determining . . . using file metadata" clause of this limitation. *Id.*

Patent Owner, however, contends that Havewala's ghosting filter (i.e., the "software client") does not determine whether a file is virtual. Prelim. Resp. 14. Instead, according to Patent Owner, that determination is made by the file system itself. *Id.* Furthermore, Patent Owner contends that the

determination of whether the file is virtual does not involve the use of file metadata. *Id.* at 14–15. Instead, this determination is made based on whether the requested file’s data has been removed. *Id.* Thus, the ghosting filter learns that a file has been “ghosted” not by analyzing any file metadata, but simply by receiving an error message from the file system. *Id.* at 15. Lastly, Patent Owner observes that to the extent the ghosting filter checks the file’s metadata to learn where the file is located, that occurs only after the file system has already determined whether the file is virtual. *Id.* We agree with Patent Owner.

(1) Havewala does not use file metadata to determine whether files are virtual.

Fundamentally, Petitioner does not adequately explain how Havewala determines whether a given file is a virtual file corresponding to a physical file stored elsewhere. Pet. 37–39 (focusing on how Havewala uses metadata to determine the *location* of requested files) Instead, Petitioner argues about whether the *location* of a file is determined by examining the file’s metadata. *Id.* (section header stating “Determining Location of Files via Metadata”); *accord* Prelim. Resp. 15.

The claim limitation, however, does not recite that the *location* of a file is determined by examining the file’s metadata. Instead, this limitation recites that a determination is made as to whether the file represented in the file system on the first device is a virtual file of a corresponding file physically stored on a second device (i.e., a ghosted file). Ex. 1001, 11:36–39. Petitioner’s failure to address the particular language of the claim represents another flaw in its analysis of claim 1.

We additionally observe that Havewala does not use any file metadata when determining whether a file is a virtual file corresponding to a physical

file located elsewhere. Instead, as discussed above, Havewala teaches that its file system simply attempts to access the requested file using its standard methods. Ex. 1005 ¶¶ 64, 68. That request either succeeds or fails.

A request fails when a requested file does not contain the requested data (i.e., the file has been ghosted), which causes file system 16 to return an error message. Ex. 1005 ¶ 68; *see also id.* ¶ 64. Then, “the ghosting filter 26 intercepts such returned error and perceives *based thereon* that the file 14 at issue is in fact in a ghosted format (step 605).” *Id.* ¶ 68 (emphasis added). Thus, Havewala’s determination that a file is “a virtual file corresponding to a physical file stored on a second device” as claimed is based on the receipt of a returned error message, not on any file metadata.⁸

In view of the above, we are not persuaded that Havewala makes any determination of whether a file is a virtual file “by reviewing file metadata” as claimed. Thus, Petitioner has not shown sufficiently that Havewala teaches this limitation.

(2) *Havewala does not teach using file metadata to determine a file’s location based on a request for that file.*

Even were we to be persuaded (we are not) that this limitation requires determining the location of files using file metadata, as Petitioner

⁸ For completeness, we observe that a request succeeds when the requested file is physically located on the first device, but this operation does not use file metadata either. The request is simply processed and the file is opened as normal. *Cf.* Ex. 1005 ¶ 68 (explaining that for ghosted files, the ghosting filter creates a handle corresponding to the ghosted file “and delivers the handle to the application 30 as the (normal) response to the open command therefrom (step 609).”). Thus, in this situation, Havewala’s determination that a requested file is a physical file is based on the fact that the requested file contained the data 20, and not based on any file metadata.

contends, Petitioner has not shown sufficiently that Havewala teaches this feature.

(a) *“Open” requests are not intercepted by the client software, and thus cannot be used to satisfy this limitation.*

As discussed above, Havewala discloses two types of file requests, “open” and “read” requests. Havewala’s “open” requests are not intercepted by the client software, and thus do not teach the “request” recited in Claim 1. *Supra*, Section II.C.4.b.(1).

(b) *Petitioner fails to explain adequately how “read” requests use file metadata.*

Havewala further discloses “read” requests, which are sent by applications or the operating system to the file system once a file has been successfully opened. Ex. 1005 ¶¶ 69–71. As noted above, Petitioner only mentions these requests in passing without adequate explanation, and thus fails to establish that the “read” requests are intercepted by Havewala’s ghosting filter. Petitioner also has not adequately explained if or how the ghosting filter uses file metadata to determine whether the file being read is a virtual file corresponding to a physical file stored on a second device” as claimed.⁹

Therefore, Petitioner has not shown sufficiently that Havewala teaches this feature.

⁹ We observe that Havewala’s discussion of reconstituting a ghosted file in response to a “read” request states that “the ghosting filter 26 issues to the ghosting manager 28 a request . . . where such request includes the ID from the ghosting information 24 of the ghosted file 14.” Ex. 1005 ¶ 73. Petitioner, however, provides no discussion of this passage of Havewala, much less any explanation of how, if at all, this passage bears on this limitation.

d) Conclusion as to Claim 1

Based on the present record, Petitioner has not shown sufficiently that Havewala teaches at least the following limitations of claim 1:

1. “intercepting the request by a software client on the first device;”
and
2. “determining by the software client if the file is physically located on the first device or if the file is a virtual file of a corresponding file physically stored on a second device by reviewing file metadata.”

Petitioner does not identify any teachings from Adams that bear on these limitations. Therefore, Petitioner does not demonstrate a reasonable likelihood of success that it would prevail in showing that claim 1 is unpatentable.

5. Analysis of Claim 7

Claim 7 is a *Beauregard* claim (*see In re Beauregard*, 53 F.3d 1583 (Fed. Cir. 1995)) that recites the identical process set forth in claim 1, fixed in a non-transitory computer-readable medium. *Compare* Ex. 1001, 12:5–35, *with id.* at 11:25–52.

For the similarly-recited limitations of claim 7, Petitioner asserts the same arguments as for claim 1. Pet. 64–66. For the same reasons set forth above for claim 1, we are not persuaded that Havewala in view of Adams teaches all limitations of claim 7. Therefore, Petitioner does not demonstrate a reasonable likelihood of success that it would prevail in showing that claim 7 is unpatentable.

6. Analysis of Claim 8

Claim 8 recites a client having a memory and a processor. Ex. 1001, 12:36–38. The processor is configured to perform the same process set forth

in claim 1. *Compare* Ex. 1001, 12:36–65, *with id.* at 11:25–52. For the remaining limitations of claim 8, Petitioner asserts the same arguments as for claim 1. Pet. 66–68. For the same reasons set forth above for claim 1, we are not persuaded that Havewala in view of Adams teaches all limitations of claim 8. Therefore, Petitioner does not demonstrate a reasonable likelihood of success that it would prevail in showing that claim 8 is unpatentable.

D. Havewala in view of Adams and Saridakis: Claims 2 and 3 (Ground 2)

Claims 2 and 3 depend from claim 1. As set forth above for Ground 1, Petitioner does not demonstrate a reasonable likelihood of success in showing that claim 1 is unpatentable because of the deficiencies in Petitioner’s analysis of Havewala discussed above. Petitioner relies on Saridakis solely to address limitations found in dependent claims 2 and 3. Pet. 68–71. Therefore, Petitioner’s citations to Saridakis do not cure the deficiencies noted above for Havewala. For these reasons, Petitioner does not demonstrate a reasonable likelihood of success that it would prevail in showing that claims 2 and 3 are unpatentable over the proposed combination of Havewala, Adams, and Saridakis.

E. Ground 3: Havewala in view of Adams, Saridakis, and Rothman (claims 1–8)

Petitioner relies in the alternative on Rothman to teach a “singular file system” as that term is used in claims 1, 7, and 8. Otherwise, this ground presents the same arguments as in Grounds 1 and 2. *Compare* Pet. 11–55, 68–70, *with id.* at 73–77. Accordingly, for the same reasons set forth above for Grounds 1 and 2, we are not persuaded that Havewala in view of Adams, Saridaiks, and Rothman teaches all limitations of any challenged claim. Therefore, Petitioner does not demonstrate a reasonable likelihood of

success that it would prevail in showing that claims 1–8 are unpatentable on this Ground.

III. CONCLUSION

For the foregoing reasons, we determine that Petitioner has not shown a reasonable likelihood that it would prevail in showing that at least one of the challenged claims of the '260 Patent is unpatentable.

IV. ORDER

In consideration of the foregoing, it is hereby:

ORDERED that no *inter partes* review is instituted in this proceeding.

IPR2024-00285
Patent 8,484,260 B2

FOR PETITIONER:

Dion M. Bregman
Michael J. Lyons
Austin L. Zuck
MORGAN, LEWIS & BOCKIUS LLP
dion.bregman@morganlewis.com
michael.lyons@morganlewis.com
austin.zuck@morganlewis.com

FOR PATENT OWNER:

Rex Hwang
John E. Lord
Alexander E. Gasser
SKIERMONT DERBY LLP
rhwang@skiermontderby.com
jlord@onellp.com
agasser@skiermontderby.com