

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

DROPBOX, INC. AND MICROSOFT CORPORATION,
Petitioners,

v.

DATANET LLC,
Patent Owner

Case IPR2024-00079
Patent 9,218,348

**PATENT OWNER'S PRELIMINARY RESPONSE UNDER 37
C.F.R. § 42.107 TO PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 9,218,348**

Mail Stop "PATENT BOARD"
Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

EXHIBIT LIST

Exhibit	Description
2001	DECLARATION OF ZAYDOON (“JAY”) JAWADI IN SUPPORT OF PATENT OWNER’S PRELIMINARY RESPONSE
2002	Curriculum Vitae of Zaydoon (“Jay”) Jawadi
2003	Computer Busses, Bill Buchanan, CRC Press, 2000, pp. 139-143
2004	The Multi-boot Configuration Handbook, Roderick W. Smith, Que, 2000, p. 71, 76-79
2005	Encyclopedia of Microcomputers, Volume 2 - Authoring Systems for Interactive Video to Compiler Design; Allen Kent, James G. Williams, and Rosalind Kent; Taylor & Francis; 1988; pp. 196-197
2006	PhoenixBIOS 4.0 Revision 6 User's Manual; Phoenix Technologies Ltd.; June 22, 2000; pp. ii-iv, 31-34, 39, 43-51, 80
2007	Forensic Computing: A Practitioner's Guide; A. J. Sammes, Brian Jenkinson; Springer Science & Business Media; 2000; pp. 119; 132
2008	Introduction to Software Engineering; Ronald J. Leach; CRC-Press; 2000; p. 292
2009	PC Interrupts: A Programmer's Reference to BIOS, DOS, and Third-party Calls; Ralf Brown, James Kyle; Addison-Wesley; 1991; pp. 7-8, 205-211
2010	Hard Disk Controller: the Disk Drive’s Brain and Body; James Jeppesen, Walt Allen, Steve Anderson, Michael Pilsl; Infineon Technologies; 2001
2011	SCSI-2 Standard; ANSI; 1994; pp. i-vi, 150-217
2012	ATA Draft Standard; ANSI; 1994; pp. i-iii, 33, 52-209
2013	Newton's Telecom Dictionary: The Official Dictionary of Telecommunications Networking and Internet; 16th Edition; Harry Newton, Ray Horak; Cmp Books; 2000; p. 398
2014	U.S. Patent Publication No. 2009/0254594 to Laurion D. Burchall
2015	<i>Reserved</i>
2016	U.S. Patent Publication No. US20070226436 to Cheng et al.
2017	U.S. Patent Publication No. US20040003103 to Witt et al.
2018	Foundations of Computer Technology; A. John Anderson; 1994; Taylor & Francis; pp. 277, 282, 288-290
2019	Redline comparing Pet., 13-18 with EX1003 ¶¶49-58

TABLE OF CONTENTS

I.	Introduction.....	1
II.	Background.....	2
A.	The '348 Patent Invention	2
B.	Level of Ordinary Skill in the Art.....	6
C.	Petitioner's references.....	6
1.	Kenley [EX1014]	7
2.	Long [EX1004]	9
3.	Cabrera [EX1005]	11
4.	Schneider [EX1016].....	11
5.	Whiting [EX1006].....	12
III.	Legal Standards.....	12
IV.	Claim Construction	14
V.	Argument	15
A.	Ground 1: Kenley+Long.....	15
1.	The Petition fails to show that a POSITA would have combined Kenley and Long	15
B.	Ground 2: Cabrera+Schneider+Whiting.....	39
1.	Petitioner's Claim Mapping is Material Deficient.....	39
2.	Petitioner failed to meet its burden in establishing “(a) the step of detecting an instruction by a resident program in a computing device for performing an operation on an operating file;” claim elements [1.2], [8.2], [12.2], [13.2], [14.2], and [15.2].....	39
3.	Petitioner failed to meet its burden in establishing, “(b) the step of creating an archive file from the operating file and storing the archive file in a temporary storage location temporally proximate to the operation being performed on the operating file and responsive to detecting the instruction; ;” claim elements [1.3], [8.3], [12.3], [13.3], [14.3], and [15.3].....	48
4.	Petitioner failed to meet its burden in establishing, “identifying presence of the archive file in the temporary storage location responsive to the occurrence of a first event” claim elements [1.4], [8.4], [12.4], [13.4], [14.4], and [15.4].. ..	52

5. No Motivation to combine Cabrera-Schneider-Whiting in the claimed manner.....56

VI. Conclusion 65

I. Introduction

Petitioners Dropbox, Inc. and Microsoft Corporation (collectively “Petitioner”) have challenged Claims 1-6, 8-20, and 23-31 (the “Challenged Claims”) of U.S. Patent No. 9,218,348 (the “’348 Patent”). Patent Owner Datanet LLC (“Datanet” or “Patent Owner”) respectfully requests that the Board decline to institute *inter partes* review of the Challenged Claims because Petitioner has not shown a reasonable likelihood of prevailing on the invalidity grounds presented for the Challenged Claims.

For the purposes of this Preliminary Response, Patent Owner addresses only the independent claims challenged by Petitioner, and only certain deficiencies in Petitioner’s grounds common to all Challenged Claims.¹ Because Petitioner has failed to demonstrate a reasonable likelihood of success of invalidating any of the independent claims (and by extension any of the dependent claims) in light of the arguments below, the rest of its challenges necessarily fail. For the following reasons, the Petition should be denied.

¹ Patent Owner will offer additional arguments and evidence addressing all Challenged Claims and grounds if the Petition is instituted.

II. Background

A. The '348 Patent Invention

The '348 Patent is titled “AUTOMATIC REAL-TIME FILE MANAGEMENT METHOD AND APPARATUS.” EX1001. The patent was filed on June 24, 2013, issued on Dec. 22, 2015, and claims priority to Provisional Application No. 60/234,221 (the “'221 Provisional”), filed September 21, 2000. EX1001, 1:4-6.

The '348 Patent relates to a method for archiving files. *See, e.g.*, EX1001, Claim 1 and Abstract. The Patent describes a novel method of backing up a computer—providing real-time, continuous backup even when the desired backup location is not available, and doing so with an imperceptible impact on system performance from a user’s point of view.

The '348 Patent is directed towards backup technology dating back to the late 1990s and early 2000s. During that time, organizations and companies struggled to provide their users with satisfactory data backup solutions. *See* EX2001, ¶¶76-89. In fact, system administrators typically approached the problem of data backup by configuring their servers and computer systems to regularly run programs that would attempt to copy files from a user’s home directory to a network storage device. Such programs were run according to a predetermined fixed schedule and due to slow network data transmission rates and limited storage availability, the backup ‘jobs’

would typically be run at most every hour, but more likely once a day or once per week. *See* EX1001, 2:17–30.

Because backups were executed according to a predetermined and fixed schedule, file modifications made by the user since the last backup operation would be lost if the computer crashed or the file was lost or corrupted. EX1001, 2:13–30. Moreover, even running backups once per hour could lead to significant data losses and these conventional approaches required that the backup storage device be connected and available at the time of the scheduled backup. EX1001, 2:13–30. In addition, if the backup storage device was unavailable, the scheduled backup would fail and a further backup would not be attempted until the next scheduled backup operation. EX1001, 2:17–30. Outdated backup files and data loss were the norm, with very real consequences. “As reported by the Safeware Insurance Agency, in 1999 alone, insurance claims for damaged, lost and stolen computers (primarily notebook computers) totaled more than \$1.9 billion. This figure does not include the untold billions lost in intellectual capital and time.” EX1001, 1:48–53.

The ‘348 Patent claims are directed towards, *inter alia*, “[a] method for archiving files” EX1001, 8:45. The inventors developed an entirely new way of managing data protection and backups that fundamentally altered the way computer backups operated, providing near real-time backup operations even when backup storage devices are temporarily unavailable. EX1001, 6:39–48. Instead of relying

upon scheduled backups, Datanet's technology actively monitored a computer's operating system for, as one example, requests to save a modified file. EX1001, 7:16–7:32. Upon detecting those operating system activities, the technology initiates, in near real time, a backup operation to a local queue or buffer. *Id.* When a suitable backup storage device becomes available, the modified file contents are transmitted to the backup storage device for long-term storage. EX1001, 7:16-7:32.

By way of illustrative example only, the real-time nature of the backup solution of the '348 Patent is provided with reference to FIG. 3 (reproduced below).

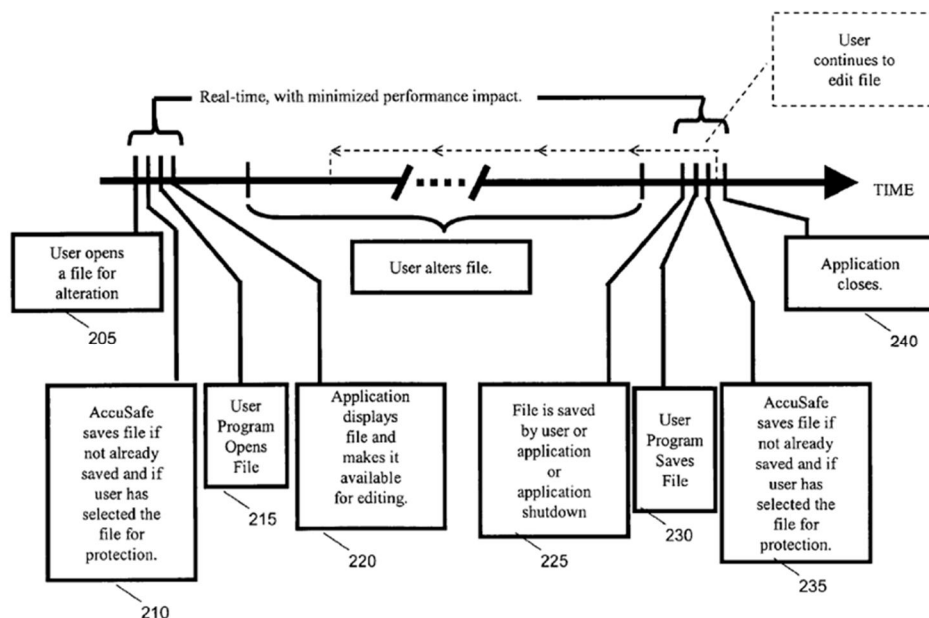


FIGURE 3

Referring to FIG. 3, the user's initial request to open a file is detected (step 205) by monitoring the operating system's file system access operations. Upon detecting the request to open a file (e.g., to make a change to that file) an immediate backup of the file in its original state is generated (step 210). EX1001, 6:60-7:32.

After the user is finished working and the user saves the file (step 225), triggering an additional operating system file system access operation, the operating system's file save operations are detected and an additional backup can be created to capture the user's modifications (step 235). *Id.*

The backup approach illustrated in FIG. 3 of the '348 Patent, therefore, is one in which contents of a working file are captured and archived at a time temporally proximate to a computer's operating system attempting to access the file (e.g., to modify the file's contents)

The claim language of Claim 1 is reproduced below:

[1.1]	A method for archiving files, comprising steps of (a) to (i) following:
[1.2]	(a) the step of detecting an instruction by a resident program in a computing device for performing an operation on an operating file;
[1.3]	(b) the step of creating an archive file from the operating file and storing the archive file in a temporary storage location temporally proximate to the operation being performed on the operating file and responsive to detecting the instruction;
[1.4]	(c) the step of identifying presence of the archive file in the temporary storage location responsive to the occurrence of a first event;
[1.5]	(d) the step of transmitting the archive file to a second storage location responsive to a second event, the second storage location being an intermediate or a permanent storage location;
[1.6]	(e) after storing the archive file in the temporary storage location, updating a database to indicate that the archive file is located in the temporary storage location;
[1.7]	(f) determining a final destination for the archive file;
[1.8]	(g) moving the archive file from the temporary storage location to an intermediate storage location;
[1.9]	(h) updating the database to indicate the archive file is located in the intermediate storage location; and

[1.10]	(i) after moving the archive file to the second storage location, updating the database to indicate the archive file is located in the second storage location.
--------	---

'348 Patent Claim 1 at 8:45-9:5.

B. Level of Ordinary Skill in the Art

Petitioner asserts that a POSITA “would have held a bachelor’s degree in computer science, electrical engineering, computer engineering, or a related field, and one or more years of experience in the design and development of file management, restoration, and archival systems.” Pet., 6-7. Patent Owner does not offer a competing definition at this time, because even under Petitioner’s proposed definition the Petition fails to present a reasonable likelihood of success.²

C. Petitioner’s references

Petitioner’s invalidity grounds are based on U.S. Patent No. 5,276,867 (“Kenley”), U.S. Patent No. 5,325,519 (“Long”), U.S. Patent No. 5,953,729 (“Cabrera”), PCT Publication No. WO 99/12101 (“Schneider”), and U.S. Patent No. 5,778,395 (“Whiting”). Petitioner relies on two obviousness grounds:

Ground	Claims	References
1	1-6, 8-20, 23-31	Kenley+Long
2	1-6, 8-20, 23-31	Cabrera+Schneider+Whiting

Pet., 2.

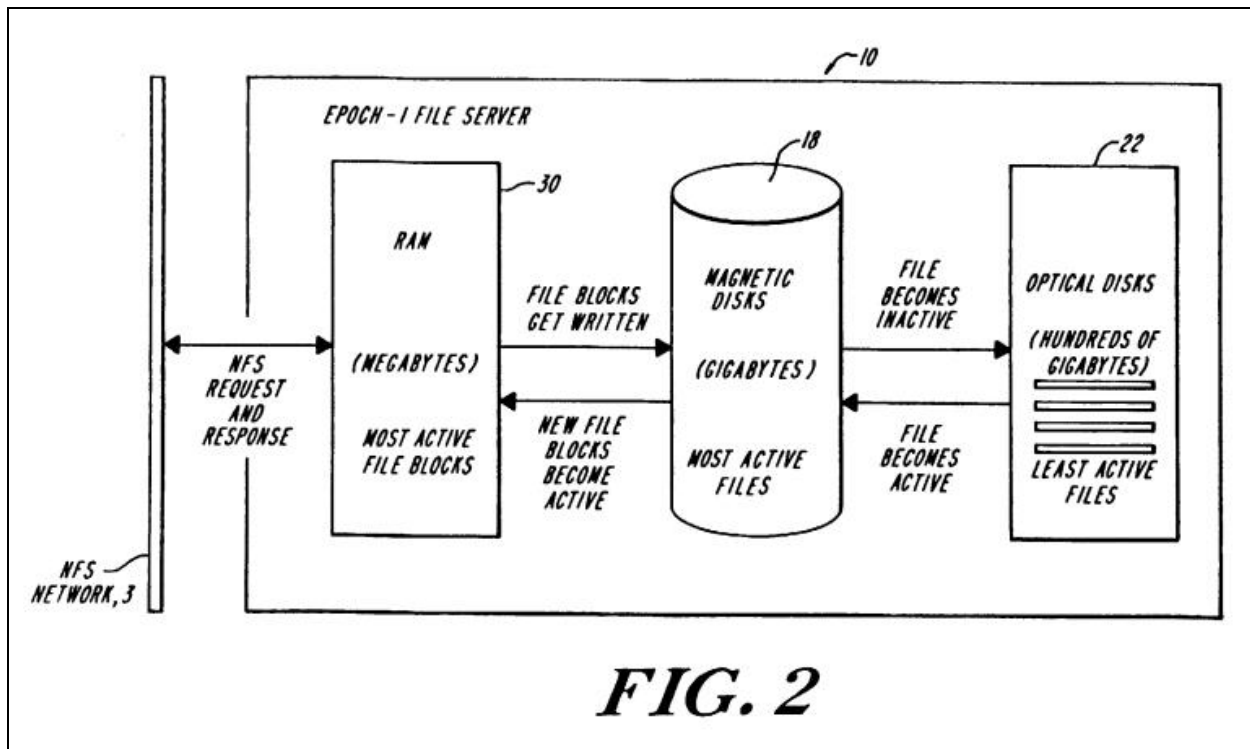
² Patent Owner reserves the right to propose a competing definition if the Petition is instituted.

1. Kenley [EX1014]

Kenley is titled “Digital Data Storage System With Improved Data Migration.” EX1014 at 1. Kenley teaches a variation on the concept of hierarchical storage system for files, noting a “hierarchical” storage server “emulates a large amount of high-speed secondary storage through use of a fixed amount of high-speed secondary storage as a cache for a large amount of slower tertiary storage.” *Id.*, 4:48-51; EX2001, ¶166. “This technique may be applied recursively to an arbitrary depth, hence the name ‘hierarchical’ storage server.” *Id.*, 4:51-53.

Kenley purports to provide “a hierarchical digital data storage system enabling storage and ‘transparent’ access to digital data.” EX1014, 2:33-36. The stated goal for the system is “provid[ing] digital data storage systems having the speed and convenience of online data storage, and the economy and large capacity of offline storage.” EX1014, 2:19-22.

Kenley gives Fig. 2 as the example of “the structure of a multiple-level storage hierarchy utilized in accordance with the invention”:



EX1014, Fig. 2, 4:32-34. The structure “includes RAM 30, magnetic disks 18, and optical disks 22.” EX1014, 13:49-50. RAM is used to service the “Most Active File Blocks,” while “relatively inexpensive optical volumes are used to emulate large magnetic disk capacities” and “[t]he high speed magnetic disks 18 act as a cache for the most recently accessed files, while less active files are kept on removable optical diskettes 22.” *Id.*, 13:53-57.

Kenley suggests using “a primary storage element, a secondary storage element, and a backing store element, each selectively storing and providing access to first, second, and third sets of data, respectively.” *Id.*, 2:44-48. “[T]he backing store stores data organized as files, and includes a file access element enabling a user or an applications program to request selected files stored in backing store. The data

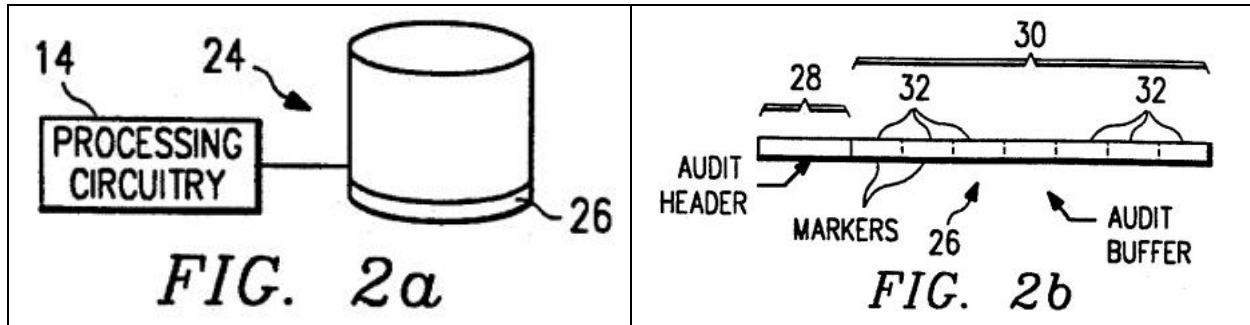
file access element includes a data file stage-in element for migrating, or ‘staging-in’ selected data files stored on the backing store to the primary storage element, and concurrently writing the staged-in data files to the secondary storage element for future access.” *Id.*, 3:29-34. Kenley is completely dependent on a file system level implementation, *id.*, 5:16-20, and works by staging-in and staging-out particular files through the various storage tiers, *id.*, 5:10-43.

Kenley and its cache based hierarchical approach to storing files has little relevance to the file archiving Claims of the ’348 Patent beyond the fact that both deal with files. Petitioner relies primarily on Kenley’s discussion of files for purposes of its Ground 1 challenge.

2. Long [EX1004]

Long is titled “Fault Tolerant Computer With Archival Rollback Capabilities.” EX1004, 1. Long relies on capturing images of hard disk sectors for purposes of hard disk error recovery. EX1004, 1:56-59 (“Therefore, a need has arisen in the industry for a fault tolerant system having an effective and cost efficient method of recovering from an error affecting the hard disk drive.”), 7:37-40 (The data to be modified on the disk is read one sector at a time (block 50). Each sector is read from the disk (or from a disk cache, if provided) and is placed in the audit buffer as ‘before image’ data (block 52).”, Fig. 5; *See* EX2001, ¶¶128-165.

Long Figures 2a and 2b “illustrate a diagrammatic representations of allocation of the hard disk and the audit partition” relied on by Long:



Id., Figs. 2a-2b, 2:23-24.

Fig. 2a illustrates the invention’s hard disk 24, “having an audit partition region 26.” *Id.* at 4:28-39. “As shown in FIG.2b, the audit partition 26 is used to keep track of the audit environment and to store certain selected events that occur within the system.” *Id.* at 4:40-42. “[T]he audit partition is structured as series of disk sectors; the first disk sector contains the audit header 28 and the following sectors contain the audit buffer 30.” *Id.* at 4:53-56. “The audit header is comprised of a number of fields, defined in Table 1.” *Id.* at 4:57-58. As can be seen in Table 1, these fields are specific to the hard disk level, including information such as the location of the audit buffer on disk (and in the disk cache) and several sector specific items. *See Id.* at 4:61-5:25 (discussing fields such as SeqNum, BufLoc, CacheLoc, MaxSectors, and CurSectors).

Long’s teachings are a variation of the long-recognized concept of using hard disk drive imaging to restore a hard drive after an error or failure. *See EX1004*, 3:1-

50; EX2001, ¶¶82-89, 128-165. It teaches that error detection and containment should be “confined at the lowest possible level” and that errors at that level “are more easily detected than errors originating within modules that generate or transform data.” EX1004, 3:1-50. Long never addresses the concept of “files” or “file backup”—the word “file” never appears in the disclosure. The low (near the hardware) level implementation of Long could not (and would not need to) even recognize the concept of files.

3. Cabrera [EX1005]

Cabrera is titled “Using Sparse File Technology To Stage Data That Will Then Be Stored In Remote Storage.” EX1005 at 1. Cabrera teaches archiving or backing up data produced from one or more data producers that is copied from local storage to the staging area and stored there temporarily until it is transferred to backup or archive storage. EX1005, Abs., Fig. 2, 6:58-8:49.

4. Schneider [EX1020]

Schneider is titled “Using Sparse File Technology To Stage Data That Will Then Be Stored In Remote Storage.” EX1005 at 1. Schneider discloses reverting a disk drive to an earlier point in time. EX1020, Abs. Schneider teaches, “changes made to the drive are saved in a circular history buffer which includes the old data, the time it was replaced by new data, and the original location of the data.” *Id.* “The circular history buffer may also be implemented by saving new data elements into

new locations and leaving the old data elements in their original locations.” *Id.* “The disk drive is reverted to an earlier point in time by replacing the new data element with the original data elements retrieved from the history buffer” *Id.*

Schnieder teaches a separate and alternative file level embodiment, noting “[a]lthough the present invention at the file level is possible it may be more difficult to implement, prove correct, and isolate as a subsystem.” EX1020, 24:20-22. “Such an implementation could save an entire prior file state just before a file is opened and the intention to modify is clear, or it could attempt to only save prior states of the portion of the file being modified.” EX1020, 24:20-24. A sector level approach is decoupled or independent from an operating system that can prevent catastrophic information data loss that can happen in a file level approach. EX1020, 10:28-11:4, 13:14-17, 14:4-9, and EX2001, ¶¶318-323.

5. Whiting [EX1006]

Whiting is titled “System For Backing Up Files From Disk Volumes On Multiple Node Of A Computer Network.” EX1006 at 1. Whiting teaches a system for backing up files from disk volumes on multipole nodes of a computer network to a common random-access backup storage means. EX1006, Abs.

III. Legal Standards

An IPR should not be instituted unless the Petitioner has shown a likelihood of success on the invalidity grounds presented *in the petition*. See *In re Magnum Oil*

Tools Int’l, Ltd., 829 F.3d 1364, 1381 (Fed. Cir. 2016) (“[T]he Board must base its decision on arguments that were advanced by a party, and to which the opposing party was given a chance to respond.”). “In an IPR, the petitioner has the burden from the onset to show with particularity why the patent it challenges is unpatentable.” *Harmonic Inc. v. Avid Tech., Inc.*, 815 F.3d 1356, 1363 (Fed. Cir. 2016). This burden of persuasion never shifts to the patent owner. *See Dynamic Drinkware, LLC v. Nat’l Graphics, Inc.*, 800 F.3d 1375, 1378 (Fed. Cir. 2015). “To satisfy its burden of proving obviousness, a petitioner cannot employ mere conclusory statements. The petitioner must instead articulate specific reasoning, based on evidence of record, to support the legal conclusion of obviousness.” *In re Magnum Oil Tools Int’l*, 829 F.3d at 1380. “Most if not all inventions arise from a combination of old elements...However, identification in the prior art of each individual part claimed is insufficient to defeat patentability of the whole claimed invention. Rather, to establish obviousness based on a combination of the elements disclosed in the prior art, there must be some motivation, suggestion or teaching of the desirability of making the specific combination that was made by the applicant.” *In re Kotzab*, 217 F.3d 1365, 1369-70 (Fed. Cir. 2000) (internal citations and quotations omitted).

35 U.S.C. § 312(a)(3) requires that a petition for *inter partes* review “identif[y]...with particularity...the evidence that supports the grounds for the

challenge to each claim.”³ The corresponding regulation requires that a petition include a “detailed explanation of the significance of the evidence.” 37 C.F.R. § 42.22(a)(2). To satisfy these requirements, a petition must, for each challenged claim, “identify...[h]ow the construed claim is unpatentable” (37 C.F.R. § 42.104(b)(4)) and “specify where each element of the claim is found in the prior art patents or printed publications relied upon.” *Id.* “A petitioner may not rely on a vague, generic, and/or meandering petition and later fault the Board for failing to understand what the petition really meant. Ultimately, it is the petitioner's burden to present a clear argument.” *Netflix, Inc. v. DivX, LLC*, 84 F.4th 1371, 1377 (Fed. Cir. 2023).

IV. Claim Construction

Claims are construed according to *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc). *See* 37 C.F.R. § 42.100(b). “Any prior claim construction determination concerning a term of the claim in a civil action...will be considered.” *Id.* Petitioner does not expressly propose any claim constructions in its Petition. Pet.,

³ Emphasis added unless otherwise noted.

7. Patent Owner does not believe any claim constructions are necessary based on the arguments presented by Patent Owner at this time.⁴

V. Argument

As detailed below, the Board should decline to institute review because neither Ground 1 nor Ground 2 presents a reasonable likelihood of success of invalidating the Challenged Claims.

A. Ground 1: Kenley+Long

1. The Petition fails to show that a POSITA would have combined Kenley and Long

Petitioner's Ground 1 challenge fails because it has not shown that a POSITA would have combined Kenley and Long in a manner that would render obvious the Challenged Claims. Petitioner does not assert that either Kenley or Long renders obvious the Challenged Claims by themselves. For example, Petitioner appears to concede that Long teaches nothing pertinent to file or file system level functionality, and thus would be unable to render obvious the Challenged Claims which all recite "method[s] for archiving files" and other file-specific limitations related to archiving files. *See* EX1001, Claims 1, 8-15 preambles.

⁴ Patent Owner may propose claim constructions to support further arguments if the Petition is instituted.

Rather, Petitioner argues that the limitations are rendered obvious by a nebulous combination consisting of “Kenley’s file-based backup system with the modification-driven archival approach taught by Long.” Pet., 13. Unsurprisingly, Petitioner dedicates little space to addressing the details of this proposed combination because a POSITA at the time of invention of the ’348 Patent would have understood that no such combination was likely to be viable (much less obvious). While Kenley and Long each discuss techniques that relate generally to the handling of data, the types of techniques relied on by each (which are still in use today) have long been recognized to be largely mutually exclusive and incompatible. *See* EX2001, ¶¶76-126. As such, Petitioner’s sparsely supported conclusions that a POSITA would have (and could have) combined the references to arrive at the Challenged Claims must be rejected.

i. Overview

As discussed above in Section II.C.2, the solution taught by Long relies on capturing images of hard disk sectors for purposes of hard disk error recovery. EX1004, 1:56-59, 7:37-40, Fig. 5; EX2001, ¶¶128-165. This is a variation of the long-recognized concept of using hard disk imaging to restore a hard drive after an error or failure. Long never addresses the concept of “files” or “file backup”. The low (near the hardware) level implementation of Long could not (and would not need to) recognize the concept of files.

Kenley, on the other hand, teaches a variation on the concept of hierarchical storage system for files. *See* Section II.C.1; EX2001, ¶166. It purports to provide “a hierarchical digital data storage system enabling storage and ‘transparent’ access to digital data.” EX1014, 2:33-36. It teaches the concept of using “a primary storage element, a secondary storage element, and a backing store element, each selectively storing and providing access to first, second, and third sets of data, respectively.” *Id.*, 2:44-48. Thus, Kenley teaches a variation on a tiered or hierarchical storage file solution. Kenley is completely dependent on a file system level implementation, *id.*, 5:16-20, and works by staging-in and staging-out particular files through the various storage tiers, *id.*, 5:48-6:18.

Importantly, file systems and physical disks/sectors exist at separate conceptual levels in data storage systems, with hardware agnostic file systems built on top of hardware specific physical disk/sector drivers. *See* EX2001, ¶¶90-94. Petitioner Microsoft confirmed this distinction between files and lower-level physical allocation units such as disk sectors in its own patent filings:

Persistent data storage exists at several levels of abstraction, ranging from higher levels that are closer to the logical view of data seen by users running application programs, to lower levels that are closer to the underlying hardware that physically implements the storage. At a higher, logical level, data is most commonly stored as files residing in volumes or partitions, which are associated with one or more hard disks. The file system, which can be regarded as a component of the operating

system executing on the computer, provides the interface between application programs and nonvolatile storage media, mapping the logically meaningful collection of data blocks in a file to their corresponding physical allocation units, or extents, located on a storage medium, such as clusters or sectors on a magnetic disk.

EX2014, ¶33.

Kenley further confirms the same understanding, explaining that a file system is “[a] subsystem of the operating system...usually constrained to a physical disk or a partition of a physical disk.” EX1014, 5:16-20.

As has long been recognized in the art, techniques and solutions for managing data at the file system level are rarely applicable to the disk/sector level, and vice versa. EX2001, ¶¶90-126. This makes sense—why would the same techniques be implemented (or even implementable) in two different levels of a data storage system, especially when one is built on top of the other? Yet, despite the well understood challenges of mixing file system and disk/sector level techniques, Petitioner’s challenge amounts to an unsupported conclusion that the mere existence and recognition of benefits of the two different conceptual systems amounts to a motivation to combine their teachings.

As discussed below, Petitioner failed to demonstrate a reasonable likelihood of success in showing that a POSITA would have combined Kenley and Long in the claimed manner, at least because (1) Petitioner’s own references (and prior

statements) teach away from the proposed combination, (2) a POSITA would not have had a reasonable expectation of success at combining the references in the proposed manner, (3) the proposed combination would change the principle of operation of the modified reference, and (4) Petitioner's purported motivations to combine the references are deficient.

ii. Teaching Away

A POSITA would not have a combined Kenley with Long (or Long with Kenley), at least because Petitioner's own references and prior statements to the USPTO teach away from such a combination. "A reference may be said to teach away when a person of ordinary skill, upon reading the reference, would be discouraged from following the path set out in the reference, or would be led in a direction divergent from the path that was taken by the applicant." *Depuy Spine, Inc. v. Medtronic Sofamor Danek, Inc.*, 567 F.3d 1314, 1327 (Fed. Cir. 2009) (internal quotes and citations omitted). Petitioner's references (along with others) discourage attempts to apply disk/sector level data management techniques with file system level techniques (and vice versa), and therefore Petitioner's proposed combination fails.

Long

A POSITA would have understood Long as teaching that detecting access to a storage device should occur at as low a level as possible. For example, Long

teaches that “[t]he detection of an error or its symptoms during the normal operation of the system is the cornerstone to fault tolerant architectures.” EX1004, 3:3-5. Long teaches that its error detection methods are for “[e]rrors occurring in [low-level] data-storage components, such as registers and memory” because they are far easier to detect than “errors originating within modules that generate or transform data” (as would be the case with file system level modules). EX1004, 3:7-11; EX2001, ¶¶132-133, 150-157, 183-187. Here, Long is specifically teaching away from trying to implement its error detection capabilities at a level used to “generate or transform data” such as the file system level at which Kenley is directed.

Furthermore, Long specifically teaches that errors must be “confined to the lowest possible level.” EX1004, 3:27-34. It teaches accomplishing this through low-level “processing circuitry [] to detect access to the mass storage device.” EX1004, 1:65-2:2. It further suggests including the audit subsystem⁵ as part of the system BIOS. EX1004, 6:59-63. As discussed by Patent Owner’s expert, file systems are implemented at a higher level than the BIOS—the BIOS would not have even recognized the existence of file systems which are implemented at the operating system level. EX2001, ¶¶120-126 (BIOS overview), 151-157 (Long BIOS

⁵ Petitioner suggests that the “audit subsystem...triggers [the] backup process.” Pet., 9.

implementation), 159-162 (same). Thus, a POSITA would have understood Long to be teaching away from implementing its techniques at a file system level.

File systems generally need to be hardware agnostic (unlike hardware specific disk/sector level logic). EX2001, ¶¶91-94, 111-117, 159-162, 287. Thus, a POSITA would have understood Long to be teaching away from incorporating its disk/sector level methods into any sort of file system implementation (or vice versa), which would necessarily need to be implemented at a much higher level than the computer's "processing circuitry" or BIOS taught by Long, and which would destroy the file system's ability to remain hardware agnostic. EX2001, ¶¶91-94, 111-117, 159-162, 287. Similarly, a POSITA looking at improving a file system level software implementation would not have looked to reference such as Long, which teaches that its teachings are most effective when implemented at the lowest possible level (something not possible with file systems). *See* EX1004, 3:1-49.

Furthermore, Long teaches that its teachings require rebooting the operating system after manipulating data before the updated data is usable by the operating system. EX1004, 9:37-50. Such techniques would be near useless to a file system level implementation which runs at the operating system level. EX2001, ¶¶149, 186; *see also id.*, ¶¶147-157.

Schneider

Petitioner's other references also teach away from combining a file system level implementation with a hard disk level implementation. For example, Schneider (EX1020) recognizes that file level backup and disk/sector level backup are separate and distinct methods of backing up data. *See* EX2001, ¶¶90-119, 168-181, 227-242. Schneider even recommends against trying to read individual files from a "sector level disk image" backup, noting that it would be inefficient to attempt because "a file is not stored continuously in one area of the disk, but may be spread out and intermixed with other files across the entire disk." EX1020, 2:23-28. Schneider teaches that backing up individual files (rather than sectors) is the preferred alternative if there is a need to "efficiently restore one or more files to a disk that may be both different and already containing data." *Id.*, 2:28-3:2, 5:8-16 (explaining that "a user may backup a specific set of files" as an alternative to "making a duplicate copy of the disks information" with a disk/sector level backup). Importantly, it recognizes that file and disk sector level implementations are not complementary techniques, but rather mutually exclusive alternatives. *See* EX2001, ¶¶168-181, 90-119.

Schneider also teaches that relying on file level backup (instead of a disk/sector level backup) can make data recovery impossible in certain circumstances—thus thwarting the purpose of the specific type of backup. For

example, Schneider teaches that if a file level backup is being used, and “the directory system for a disk...gets corrupted” then even files that were unaffected by the corruption may still be lost and unrecoverable. EX1020, 6:1-7 (“[L]osing the directory of a disk’s contents results in losing the referenced files, even though they still exist on the disk. In this case the information the user wants to restore is the disk’s directory.”). Petitioner argues that the reason for making the combination would be to “improve reliability,” yet applying Long’s techniques to a “file-based backup system” (rather than utilizing a disk/sector level backup) would yield a less reliable system incapable of recovering from corruption that Long’s disk/sector based approach would otherwise be able to remedy. *See Id.*; Pet., 13.

Further teaching away from the combination, Schneider teaches that another “reason why it is not desirable to incorporate the saving of original [disk] states in the operating system is complexity and bugs. **Intertwining an operating system’s filing system with the old state mapping leads to bugs that can corrupt everything.** However, by separating out the management of the original old disk states from the operating system, this relatively simple management system could recover from the operating system's filing system bugs.” EX1020, 14:4-9. Thus, Schneider recognizes that file-based backup systems and disk/sector based backup systems are largely mutually exclusive—attempting to apply techniques from one to the other results in undesirable outcomes. EX2001, ¶¶318-319.

Petitioner's Patent Filings

Petitioner Microsoft's own patent files further teach away from the proposed combination. For example, Petitioner Microsoft notes in the background section of one of its filings that “disk imaging” and “file backup” solutions were known in the art and both still widely used at the time of filing (Feb. 21, 2006)—well after the time that it supposedly would have been obvious to combine them. *See* EX2016, ¶¶1-2; EX2001, ¶¶270-275. The reasons both were still in use at the time is because each approach entailed limitations inherent to their use—limitations which prevented any obvious combination. For example, disk imaging/hard disk sector solutions could not differentiate useful data from non-useful data, and as a result, they made inefficient use of both storage and network resources. EX2016, ¶1. On the other hand, file backup solutions (which required an operating system to function), could not capture operating system files or preserve non-file information such as that related to disk partitions and boot sectors. *Id.*, ¶2. Microsoft's patent application purports to have invented a way to modify disk images to overcome **some** of the limitations related to disk imaging. *See, e.g., Id.*, ¶¶4-5, 13 (suggesting the solution “images the logical contents of the volume 101, rather than imaging the volume 101 on a sector-by-sector basis”), Claim 12. While Microsoft's Petition **now** claims it would have been obvious to try to combine aspects of file system and disk sector-based solutions around the Sept. 2000 time frame to which the '348 Patent

claims priority, as late as 2006 it was telling the USPTO the opposite and trying to claim that solutions to that very problem were so non-obvious that they were patentable. Petitioner cannot overcome such wild swings in its representations to the USPTO with a few conclusory paragraphs of expert testimony that were largely copied verbatim from its own Petition. *See generally* EX2019 (comparing Pet., 13-18 with EX1003, ¶¶49-58); *see Xerox v. Bytemark*, IPR2022-0624, Paper 9 at 15 (P.T.A.B. Aug. 24, 2022) (precedential) (giving “little weight” to a declaration which “merely repeats, verbatim, the conclusory assertion for which it is offered to support”). Thus, Petitioner’s cited references and past statements to the USPTO teach away from the proposed combination.

iii. Reasonable Expectation of Success

Additionally, a POSITA would not have had a reasonable expectation of success in combining Kenley and Long, because a POSITA at the time would have understood that teachings specific to a disk/sector level implementation could not be reasonably expected to succeed in a file system level implementation (and vice versa). “Evidence showing there was no reasonable expectation of success may support a conclusion of nonobviousness.” MPEP 2143.02 (citing *In re Rinehart*, 531 F.2d 1048(CCPA 1976)). “Whether an art is predictable or whether the proposed modification or combination of the prior art has a reasonable expectation of success is determined at the time the invention was made for pre-AIA obviousness analysis.”

MPEP 2143.02 (citing *Ex parte Erlich*, 3 USPQ2d 1011, 1016 (Bd. Pat. App. & Inter. 1986)). “[T]o have a reasonable expectation of success, one must be motivated to do more than merely to vary all parameters or try each of numerous possible choices until one possibly arrived at a successful result.” *In re Stepan Co.*, 868 F.3d 1342, 1347 (Fed. Cir. 2017) (internal quotes and citations omitted). The references that Petitioner submitted as prior art (and Petitioner’s past representations to the USPTO) confirm that a POSITA would not have had a reasonable expectation of success in combining the teachings of Kenley and Long.

Schneider

Schneider confirms the unlikelihood of success in making the proposed combination. While Schneider purports to provide limited support for retrieving files, it does not actually provide any sort of integration between the traditional disk sector and file system backup approaches. Schneider primarily relates to “reverting a disk drive to an earlier point in time.” EX1020, Abstract. Schneider offers the ability to retrieve prior disk states by either (1) reverting the state of the whole disk and subsequently using the standard file system to retrieve the files (a purely disk sector approach), or (2) using the saved disk states to create a “simulated or virtual drive whose image is created by combining information on the original drive [] and the history buffer” and using the standard file system to retrieve the files from the simulated disk (another purely disk sector based approach). EX1020, 15:20-16:19.

Schneider also addresses separate strategies for a “file level implementation,” noting that such implementations would “significantly differ” from disk sector approaches and would need to be implemented at an entirely different level (“the operating system’s filing system” rather than “lower down in the disk I/O path”). EX1020, 24; *see also* EX2001, ¶¶180-181, 190-194, 241-242, 311-312. Schneider does not propose any solutions to combine the two approaches, because Schneider itself recognizes how difficult (and unlikely to succeed) such modifications would be. For example, Schneider notes that disk/sector based systems rely on fundamentally different assumptions than file based systems. EX1020, 19:9-10 (“An underlying assumption in the sector (versus file) based implementation of the present invention is that the disk will often be left in stable states...”). Schneider points out that even the alternate file embodiment that it suggests would be “difficult to implement, prove correct, and isolate as a subsystem.” *Id.*, 24:20-21. Thus again, Petitioner’s submitted references would have suggested to a POSITA that there was no reasonable expectation of success for combining a file system level implementation with a disk/sector level implementation.

Petitioner’s Patent Filings

Petitioner Microsoft’s patent filings further confirm that there would not have been a reasonable expectation of success in combining a file system level reference with a disk image level reference, even in 2002 (after the filing of the ’348 Patent).

See EX2017, Cover page. For example, Microsoft filed a patent for a purported “invention relat[ing] to managing image files” where “the image file is a single stream of data that is a sector by sector copy of the data contained on the source storage media volume [and]....used to restore the source storage media volume to its state at the time the image file was created.” *Id.*, ¶¶2, 5-6.

Petitioner Microsoft explained:

Because the image file is created as a sector by sector copy of the source storage media, and is stored as single data stream, **it is difficult to manipulate the image file. The operating system cannot readily read** the control information or **individual file data from the data stream**. As a result, **files in the data stream cannot be replaced with new files or edited. In order to update an image file, it is necessary to restore the image file to a storage media volume, perform updates to the storage media volume, and then create a new image file.**

EX2017, ¶7. Thus, Petitioner Microsoft knew that even years after the priority date of the '348 Patent it would not have been possible to easily extract or modify files contained within a sector-by-sector disk image (as would be required by its proposed combinations). EX2001, ¶¶191-194. Yet, despite being well aware that there would have been no reasonable expectation of success at the time of invention of the '348 Patent of “implement[ing] Kenley’s file-based backup system with the [disk sector] modification-driven archival approach taught by Long,” Petitioner provides no more

than a few conclusory paragraphs supporting such a combination. Pet., 13-18. Such a conclusory analysis cannot present a reasonable expectation of success in combining the references.

iv. Principle of Operation

Furthermore, the proposed combination of the Long and Kenley systems would necessarily change the principle of operation of the modified reference (regardless of which reference was modified), and therefore would not render obvious the Challenged Claims. “If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious.” MPEP 2143.01 (citing *In re Ratti*, 270 F.2d 810, 813 (CCPA 1959) where an obviousness challenge failed because the “suggested combination of references would require a substantial reconstruction and redesign of the elements shown in [the primary reference] as well as a change in the basic principle under which the [primary reference] construction was designed to operate.”). “[A] person of ordinary skill generally would not be motivated to modify a reference by contradicting its basic teachings...or by making it inoperable for its intended purpose.” *Univ. of Md. Biotechnology Inst. v. Presens Precision Sensing GmbH*, 711 F. App’x 1007, 1010 (Fed. Cir. 2017) (internal quotes and citations omitted).

Moving functionality from one level of a data layer (the disk/sector level) to a higher one (the file system) is unquestionably a “substantial redesign.” The same would be true of trying to move functionality in the other direction, from a higher to lower level. While Petitioner might try to argue that it is only moving Long’s “event-driven backup process,” Long’s “event-driven backup process” cannot be moved to a file system level implementation without changing the principle of operation. *See* Pet., 14; EX2001, ¶¶183-186, 195-199. Long teaches that the types of events it detects are “writes to the hard disk(s), power failure, power on and system reboot.” EX1004, 5:26-30. All of these events occur at a level well below what a file system implementation could detect—they cannot reasonably be implemented at a file system level without changing the principle of operation of a file system and incorporating all lower level logic into the higher data layer (thus negating one of the foundational principles of data storage systems).

Patent Owner’s expert further confirms that a combination of Kenley and Long in the proposed manner would have required a substantial redesign and reconstruction. EX2001, ¶¶185-194. For example, Patent Owner’s expert discusses the impact of relying on the types of events that Long handles:

186. [] Long [] detects various types of events, such as “writes to the hard disk(s), power failure, power on and system reboot.” These events occur at much lower layer than the operating system and file system

layer. Detecting such events is inappropriate and possibly unimplementable at the file level.

EX2001 ¶186. Thus, Petitioner's own references once again support the conclusion that a POSITA would not have combined Kenley and Long.

v. Petitioner's Deficient Motivations to Combine

Furthermore, Petitioner's asserted motivations to combine the reference are deficient. The deficiencies in the proposed combination are underscored by the fact that Petitioner never even identified which was the primary and which was the secondary reference, and instead appears to cherry pick from each as is convenient without really addressing how a proper combination could be accomplished. In light of the defects discussed above, Petitioner could not have proposed plausible specific modifications as required, and instead relies on generic statements such as (1) "A POSITA would have been motivated...to implement Kenley's file-based backup system with the modification-driven archival approach taught by Long," (suggesting Kenley should be modified as the primary reference) and (2) "When used to implement a file-based system such as Kenley's, the Long process archives files" (suggesting Long should be modified as the primary reference). Pet., 13, 19. As discussed above, a POSITA would have disregarded both proposals at least because the references at the time of invention (and even after it) taught away from such a combination, a POSITA would not have had a reasonable expectation of success in

making any of the proposed combinations, and any proposed combination would have changed the principle of operation of the modified reference.

Note that Patent Owner is not suggesting that bodily incorporation of elements of Kenley and Long would be required. Rather, the extensive evidence (from Petitioner's own references and past statements) makes clear that a POSITA would not have been motivated to make the proposed combination at least because of the references teaching away, a lack of expectation of success, and changes in the principle of operation that would have been necessary. All of these would have been well recognized by a POSITA at the time of invention of the '348 Patent.

Even ignoring the fact that Petitioner relies on a facially invalid obviousness combination due to the failure to even identify a consistent primary reference (as well as the extensive evidence that a POSITA would not have made the proposed combination), the challenge would still fail because Petitioner's proposed reasons for the combination are deficient. Rather than discuss any motivation or proposed combination in detail, Petitioner instead relied on a shotgun approach, providing multiple conclusions with most support coming solely from its expert, who largely parrots the same conclusions from Petitioner's brief. *See* EX2019 (comparing Pet., 13-18 with EX1003, ¶¶49-58).

Petitioner argues that an unspecified combination of Kenley and Long "serves the goal of reduc[ing] the time associated with conventional data migration schemes

taught by Kenley, and provides an efficient, reliable and cost[-]effective architecture for disk fault tolerance as taught by Long.” Pet., 13-14 (internal quotes and citations omitted). While Petitioner suggests that Long “provides ‘significant improvements in hard disk error recovery” by “creating backup data after each modification,” as discussed above detecting and confining hard disk errors in the manner taught by Long cannot be imported to the file system level. *Id.*; see EX2001, ¶¶147-150. Furthermore, using Long’s method to “create backup data after each modification” would quickly overwhelm any similar circular buffer approach, and would cease to even work if the hard disk was near capacity. EX2001, ¶¶276-278. As discussed above, a POSITA would not have expected that any implementation could achieve all the benefits of both a disk/sector level implementation and a file system level implementation—the two approaches are largely mutually exclusive and that is why both approaches still exist even today.

Petitioner argues that “a POSITA reading Long would immediately contemplate use of its techniques within file-based systems, for preserving data stored in files within that system.” Pet., 14. As discussed above, this is in direct conflict with Petitioner’s submitted references and prior statements to the USPTO, where Petitioner Microsoft readily acknowledged that file systems and disk/sector implementations exist at different layers and cannot be readily intermingled. *See* Sections V.A.1.ii-iv above. Notably, Petitioner’s discussion stays at the most

abstract level possible, simply stating that a POSITA would “contemplate use of” the Long approach. Pet., 14. This does not even contemplate any specific modification (because Petitioner’s theory immediately breaks down when applied to specific modifications). As a result, Petitioner can do no better than make broad generalizations about Long being used with “file-based systems” because it cannot support any actual specific use of Long in such a system. Furthermore, it does not matter that if it was known to organize using files, or that “applications in file-based systems would perform write and other operations that modified the file.” Pet., 14-15. Petitioner fails to provide a motivation to make the specific proposed modifications to its asserted references.

Petitioner also argues that “Kenley [] describes ‘stor[ing] data organized as files’ and using file attributes to track the backup location of files enables the data to be more easily and transparently recovered from archive storage media for use by an application.” Pet., 14-15 (citing EX1014, 3:29-32, 7:19-62). This once again demonstrates the weakness of its proposed combination: file attributes are only available at the file/operating system level (and thus none of the techniques of Long would be able to even read the file attributes, much less use them to “to track the backup location of files”). Like Petitioner’s other attempts to manufacture some reason to cherry pick features from each reference, this ignores the fact that many of

the features relied on simply are not compatible with techniques from other references. *See, e.g.*, EX2001, ¶¶243-300.

Petitioner then suggests that “the two references are complementary, as each teaches at least three tiers of storage.” Pet., 15. However, Petitioner cites features such as using a combination of “semiconductor memory buffer” and copying data to disk. *Id.* These same features would be found in nearly any computer-based system from the past 50 years—allowing them to provide a motivation to combine would negate all utility of the analysis. EX2001, ¶¶283-284.

Petitioner also suggests that Kenley’s recitation of “‘transparent’ access to digital data...aligns with Long’s teachings on a system that ‘allows the archiving to take place in the background.’” Pet., 15. One of the points of a file system is to provide hardware agnostic access to data. EX2001, ¶¶91-94, 111-117, 159-162, 287. Kenley purports to teach extending this hardware agnostic aspect of file systems to other storage devices. *See, e.g.*, EX1014, 3:25-28. This same type of transparency cannot be extended to the teachings of Long. Furthermore, as Patent Owner’s expert explains, there is nothing unique about the “transparency” of Kenley—it ultimately uses the same types of backup approaches (scheduling and mirroring) that were already known in the art. EX2001, ¶¶285-286.

Petitioner suggests that “Kenley’s teaching of a database-held system...dovetails with Long’s teachings on an audit header...Thus, a POSITA

would naturally augment the catalog database of Kenley to store relevant information from the audit header taught by Long, thus ‘enhanc[ing] transparent access.’” Pet., 15-16. This proposal once again ignores the details of the two references: Kenley’s system catalog relies on “file attribute data” which would not be available to Long (which operates below the file system level), and Long’s audit headers are specifically used to store disk sector information which would be useless at the higher level of the data system addressed by Kenley. *See* EX1004, 5:1-25 (Table showing sector specific data provided in Long’s audit headers); EX1014, 3:17-28 (discussing Kenley’s “system catalog element...for storing and providing transparent access to catalogued file attribute data”); EX2001, ¶¶287-289. All of these things are self-evident from the references themselves, yet Petitioner made no attempt to address them.

Petitioner also argues that “Long teaches a straightforward modification-driven archival approach that applied to known multi-tier backup-archival systems[,and] [a] POSITA would have easily applied that approach in a file-based system.” Pet., 16. Yet, as discussed above, Petitioner Microsoft has told the USPTO in the past that the opposite is true—that even after the time of invention of the ’348 Patent, it was still extremely difficult to apply disk/sector level techniques to file system level implementations. *See, e.g.*, EX2017, ¶7; EX2001, ¶¶191-194.

Petitioner also suggests “[t]he reasonable expectation of success, and overall obviousness, of implementing Long’s technique for a file-based system such as Kenley’s is confirmed by” EX1016, 46:36-47:8. Pet., 16-17. Petitioner suggests that the cited portion shows “what normally happens when data is written” and “the Long Technique (aka ‘Move’ method).” Pet., 17. But the cited portion has nothing to do with combining a disk sector level implementation with a file system level implementation—it is just an example of what is happening at the hardware level when data is written. As Patent Owner’s expert explains, the discussion of the move method does not show Long operating at a file level or even suggest that it can be implemented at the file level. *See* EX2001, ¶¶201-242. Furthermore, the disclosure that Petitioner relies on is from the *same inventors* as the EX1020 Schneider reference, which (as discussed above) clearly teaches that a POSITA would not have had a reasonable expectation of success at making the proposed change.

vi. Multiple Limitations Are Not Taught by Kenley and Long Individually

As discussed above, a POSITA would not have been motivated to combine the relevant techniques of Long and Kenley, much less in the vague ways proposed in the Petition. In the absence of any supportable motivation to combine, Petitioner’s challenge necessarily fails. For example, regarding Claim 1 Petitioner does not even cite any teachings of Kenley as being relevant to limitations 1.1 and 1.2. Pet., 19-21. Regarding limitation 1.3, Petitioner cites the recitations of the words “archive

copies” and “data files” in Kenley’s abstract, and the suggestion that “the secondary storage maintains data in the form of files” which is clearly insufficient to teach or render obvious the full limitation.⁶ Pet., 21-24.

Nor could Long render obvious the challenged claims by itself. Petitioner tellingly never directly addresses the elephant in the room regarding Long—it never mentions files or addresses any file specific data operations, and is instead directed to hardware disk/sector level techniques that are not relevant to the file specific limitations claimed in the ’348 Patent (thus, it could not render obvious any of the file related limitations). Even if Long taught detecting instructions for performing operations on disk sectors at the hardware level, that would still be insufficient to render obvious the claim limitations specific to, for example, “detecting an instruction...for performing an operation on an operating file.” Petitioner submitted no evidence to the contrary, and has instead relied on the proposition that the teachings of Long would somehow be applicable to Kenley (or vice versa), without actually addressing the extensive and well recognized difficulties in such a combination. Thus, because a POSITA would not have combined the references and

⁶ Petitioner relies on the same analysis for the first three limitations of Claims 8 and 12-15 as it did for Claim 1, and therefore its analysis is similarly deficient for those claims. Pet., 47-48.

neither reference on its own could present a reasonable likelihood of success of invalidating the Challenged Claims, Petitioner's Ground 1 necessarily fails.

B. Ground 2: Cabrera+Schneider+Whiting

Petitioner's discussion of Ground 2 consists of an incomplete claim mapping addressed as follows.

1. Petitioner's Claim Mapping is Material Deficient

Petitioner's claim mapping fails to specify where each element of the claim is found in the prior art. The mapping: (a) fails to identify how the references are applied and which reference is the primary reference, secondary reference and tertiary reference, (b) fails to address each claim element and (c) fails to explain where the claim limitations that Petitioner purports to address are found in each of the references.

The following demonstrates how Petitioner fails to satisfy its burden for some exemplary elements of each challenged independent claim—claims 1, 8, and 12-15.

2. Petitioner failed to meet its burden in establishing “(a) the step of detecting an instruction by a resident program in a computing device for performing an operation on an operating file;” claim elements [1.2], [8.2], [12.2], [13.2], [14.2], and [15.2].

Independent claims 1, 8, 12, 13, 14, and 15 recite, in pertinent part, “the step of detecting an instruction by a resident program in a computing device for performing an operation on an operating file.” ‘348 Patent, 8:47-49; (Claim 1, Element [1.2], Claim 8, Element [8.2], Claim 12, Element [12.2], Claim 13, Element

[13.2], Claim 14, Element [14.2], and Claim 15, Element [15.2]). Petitioner failed to satisfy its burden for establishing each of these features. Patent Owner notes Petitioner failed to identify with specificity which reference is the primary reference, but rather appears to assert all references are the primary by stating, “Cabrera-Schneider-Whiting.”

In the Petition the Petitioner states:

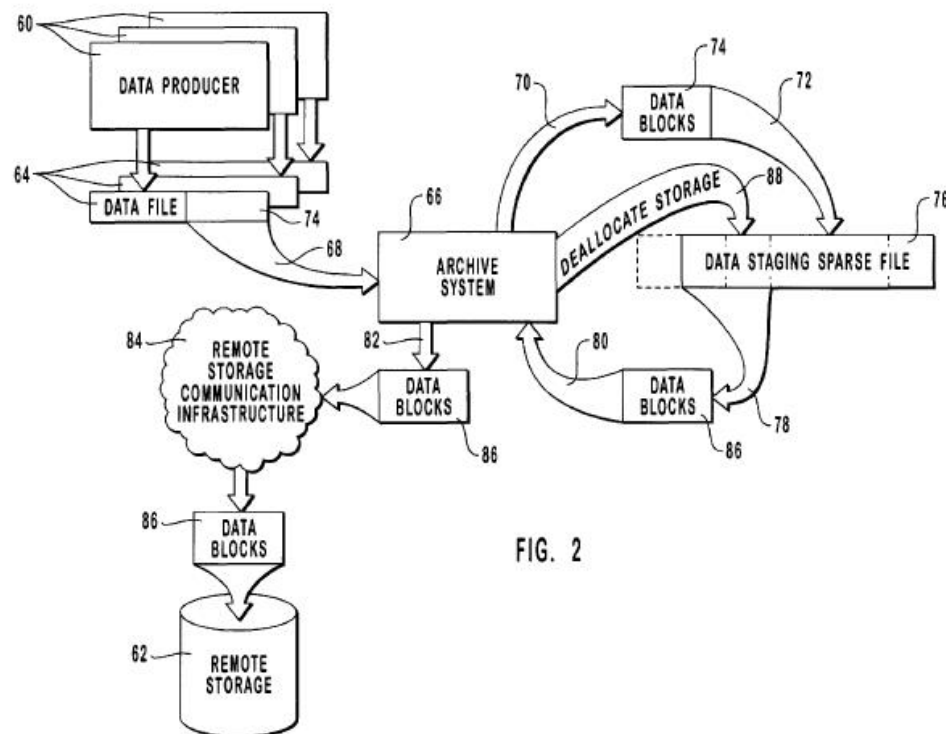
Cabrera states that “any ... triggering event” may be used to trigger copying data from local storage to the staging area. *Id.*, 10:56-61; *see also id.*, 6:58-62 (discussing creation of “data that is to be backed up or archived”); EX1003, ¶158.

Pet., 66-67.

Petitioner’s mapping of Cabrera does not identify several of the claim features including but not limited to “detecting an instruction by a resident program”, “an operation on an operating file,” or “an instruction for performing an operation on an operating file,” EX2001, ¶¶408-438. The Petition improperly shifts the burden of deciphering Petitioner’s arguments onto Patent Owner and the Board, which, as we discussed above, is contrary to the statutory and regulatory provisions governing *inter partes* review proceedings, as well as the best practices outlined in the Office Patent Trial Practice Guide. *Whole Space Industries Ltd. v. Zipshade Industrial*

(*B.V.I.) Corp.*, IPR2015-00488 Paper 14 at 18 (July 24, 2015) (citing 35 U.S.C. § 312(a)(3); 37 C.F.R. §§ 42.22(a)(2), 42.104(b)(4)–(5)).

Cabrera is directed towards an archive or backup operation performed only **after data** is present in a local storage area. Figure 2 of Cabrera is reproduced below to aid with a discussion of Cabrera.



With reference to FIG. 2, Cabrera discloses:

Referring now to FIG. 2, a high-level diagram of one embodiment of the present invention is illustrated. In FIG.2, one or more data producers, Such as data producer 60, creates data that is to be backed up or archived to a backup or archive Storage device, Such as remote Storage 62.

EX1005, 6:58-62.

Cabrera requires that after the data has already been produced from the one or more data producers it is then archived only on the occurrence of one of four events: (1) a defined amount of data has been stored locally, (2) a particular amount of time has elapsed by disclosing, (3) receipt of a command to stage data, or (4) any other triggering event. EX1005, 3:38-50; 10:56-61; EX2001, ¶¶434-438. In each of these four cases, unarchived data must already be available in the local storage, after one or more producers has finished creating the data, in order for the archive to proceed. EX2001, ¶435. No other method is disclosed in Cabrera for storing data locally other than data produced by one or more data producers. EX1005, 6:58-62. A POSITA would understand that Cabrera does not teach or suggest, “detecting an instruction by a resident program ... for performing an operation on an operating file” at least because the archiving begins only after the data resided on in the local storage. EX2001, ¶¶437-438. As noted earlier, Petitioner failed to identify the equivalent of the claimed “operating file” or what the equivalent of the claimed “operation on an operating file” is in Cabrera. Patent Owner contends that a POSITA would have understood that a triggering event cannot be “an operation on an operating file” and is also not “an instruction to perform an operation on an operating file,” at least because the triggering event occurs after the data to be archived is

already in local storage and there is no detecting at that stage in Cabrera's process. EX2001, ¶¶439-440.

Schneider fails to cure the factual deficiencies of Cabrera. A POSITA would understand that Schneider does not disclose “detecting an instruction to perform an operation on an operating file” as Schneider is directed towards at best intercepting sector-based writes and such process is completed decoupled or independent from the operating system. EX1020, 10:28-11:4, 30:4-11; EX2001, ¶¶318-323.

Petitioner asserts:

Schneider teaches, when “a request to write [to the disk] occurs, ... intercepting the write” to trigger copying the file on the disk before completing the write to the disk. EX1020, 12:5-8; *see also id.*, 30:4-6 (explaining “How to Intercept Disk Activity”). Schneider further teaches that “applications” and “an operating system (OS) ... has the responsibility ... to save and recall information from a hard disk,” and that “applications” operating under the OS generate writes “during a write cycle (e.g., the saving of a document from a word processor or the saving of a photograph from an appropriate editor).” *Id.*, 1:20-22, 32:19-20. The operating system is a resident program, and so too the applications at least because they generate writes. *See* EX1001, 4:49-52 (defining “Resident Program”); EX1003, ¶159.

Pet., 67.

Petitioner’s mapping of Schneider does not address several of the claim features including but not limited to the following features: “detecting an instruction”, “operation”, or “an operating file.” This lack of mapping improperly shifts the burden of deciphering Petitioner’s arguments onto Patent Owner and the Board, which, as discussed above, is contrary to the statutory and regulatory provisions governing *inter partes* review proceedings, as well as the best practices outlined in the Office Patent Trial Practice Guide. *Whole Space Industries Ltd. v. Zipshade Industrial (B.V.I.) Corp.*, IPR2015-00488 Paper 14 at 18 (July 24, 2015) (citing 35 U.S.C. § 312(a)(3); 37 C.F.R. §§ 42.22(a)(2), 42.104(b)(4)–(5)).

Schneider discloses:

As Writes are Generated

The history buffer works by recording the **original state of sectors** on a disk prior to being changed. The time of change is also recorded, although it is not essential, in some cases it is necessary to know the order in which changes have been made, but not the time these are made. The process is illustrated in **Figure 1**. Thus, if a **sector 10** contains value A, and a request to write a value B occurs, our method involves **intercepting the write**, reading the **sector location 10** (picking up the A value), writing this original value into the circular history buffer, and then returning to complete the write of the new B value.

EX1020, 11:30-12:8.

As shown from the foregoing, there is no discussion of a file, but rather this section is directed towards intercepting the write of a sector. EX2001, ¶¶341-343. In addition, FIG. 1 reproduced below further illustrates based on the labels above the figure and to the right of the figure that the Schneider is directed towards a sector-based write operation and not a file based write operation. EX1020, FIG. 1; EX2001, ¶¶344-347.

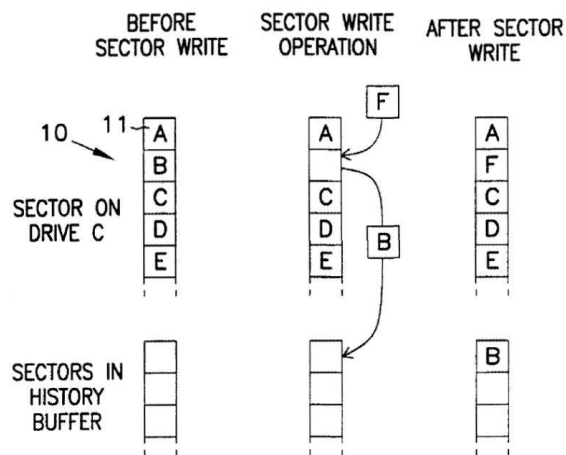


FIG. 1

EX1020, FIG. 1.

In addition, Schneider discloses:

How to Intercept Disk Activity

The technology of the present invention is based on intercepting and modifying transfers to and from a disk. The method of doing such is operating system dependent,

but typically involves writing a device driver for a hard disk that is inserted in front of that normally used. In the case of DOS the techniques are well known to accomplish this effect. See the INT 13 request. As already mentioned, a partial hardware implementation could involve putting logic of the present invention right in the disk controller.”

EX1020, 30:4-11.

A POSITA would understand that this section is also directed to sector-based write activities and **not** directed towards file activities. EX2001, ¶¶337-391. No other method is disclosed in Schneider for interception other than sector-based write operation interception. The sector-based write also has nothing to do with an operation on an operating file. EX2001, ¶¶390-391.

Schneider also teaches:

“By **implementing the disk sector embodiment** of the present invention below the operating system (as a pre-disk controller) the present invention is **de-coupled from the operating system**. Since this embodiment of the present invention **can revert a disk back to an earlier state it can recover from bugs in the operating system that might otherwise cause catastrophic information loss** by a single improper disk write. **Backup techniques that are tightly coupled with an operating system and its filing system are less able to recover from bugs in themselves**.”

EX1020, 10:28-11:4.

From at least the foregoing, a POSITA would understand that Schneider teaches the sector-based write interception sector-level functions are independent of the operating system and does not teach “detecting an instruction by an operating system” as the Petition equates an operating system with a resident program (which is not expressly addressed herein). EX2001, ¶¶321-323, emphasis added. A POSITA would also understand from the teachings of Schneider that intercepting sector-level activities is not directed towards file-level activities and the term “disk activity” is not equivalent to “file activity.” EX2001, ¶353. Schneider also teaches that file-level implementations and sector-level implementations are different, not complimentary, and largely mutually exclusive alternatives. EX1020 6:18-22; EX2001, ¶¶313-317.

In addition, a POSITA would not have relied on or even considered Schneider, because Schneider teaches away from file-level implementations and because Schneider discloses sector-based (not file-based) interception. *See* Section VB, *supra*.

The Petitioner does not reference, mention or rely upon Whiting in this section or for teaching “detecting an instruction by a resident program in a computing device for performing an operation on an operating file.” Accordingly, A POSITA would understand Cabrera-Schneider-Whiting either alone or in combination do not teach

“detecting an instruction by a resident program in a computing device for performing an operation on an operating file.”

- 3. Petitioner failed to meet its burden in establishing, “(b) the step of creating an archive file from the operating file and storing the archive file in a temporary storage location temporally proximate to the operation being performed on the operating file and responsive to detecting the instruction;” claim elements [1.3], [8.3], [12.3], [13.3], [14.3], and [15.3].**

Independent claims 1, 8, 12, 13, 14 and 15 recite, in pertinent part, “creating an archive file from the operating file and storing the archive file in a temporary storage location temporally proximate to the operation being performed on the operating file and responsive to detecting the instruction.” ‘348 Patent, 8:51-54; (Claim 1, Element [1.3], Claim 8, Element [8.3], Claim 12, Element [12.3], Claim 13, Element [13.3], Claim 14, Element [14.3], and Claim 15, Element [15.3]). Petitioner failed to satisfy its burden for establishing each of these features. Patent Owner notes, Petitioner failed to identify with specificity which reference is the primary reference, but rather appears to assert all references are the primary by stating, “Cabrera-Schneider-Whiting.”

In the Petition the Petitioner states:

Cabrera discloses copying a file “from local storage to the staging file” in response to “any ... triggering event.” EX1005, 10:56-61. The staging file or staging area “is adapted for temporarily storing the data.” Id., 7:3-7; id., 2:5-10 (“‘stage’ the information in a temporary storage

area”). Cabrera discloses that “[a]ny mechanism that performs this function [of moving data] may be utilized such as, for example, reading the data from the appropriate location and then storing a copy of the data in the staging storage area[, ...] direct memory transfer and so forth.” *Id.*, 7:11-16.

Pet., 68.

As shown, from the Petitioner’s mapping of Cabrera does not address several of the claim features including but not limited to “creating an archive file from the operating file,” “storing the archive file in a temporary storage location,” wherein the features of “creating” and “storing” are “temporally proximate to the operation being performed on the operating file,” and “responsive to detecting the instruction.” Accordingly, the Petition improperly shifts the burden of deciphering Petitioner’s arguments onto Patent Owner and the Board, which, as we discussed above, is contrary to the statutory and regulatory provisions governing *inter partes* review proceedings, as well as the best practices outlined in the Office Patent Trial Practice Guide. *Whole Space Industries Ltd. v. Zipshade Industrial (B.V.I.) Corp.*, IPR2015-00488 Paper 14 at 18 (July 24, 2015) (citing 35 U.S.C. § 312(a)(3); 37 C.F.R. §§ 42.22(a)(2), 42.104(b)(4)–(5)).

As discussed above in Section V.B.2, Cabrera further requires that **after** the **data** has already been produced from the one or more data producers, it is then archived **only** on the occurrence of one of four events: (1) a defined amount of data

has been stored locally, (2) a particular amount of time has elapsed by disclosing, (3) receipt of a command to stage data, or (4) any other triggering event. EX1005, 3:38-50; 10:56-61; EX2001, ¶¶434-437, and 445. As such, a POSITA would understand Cabrera does not teach creating the archive file from the operating file ... responsive to detecting the instruction to perform an operation on an operating file.” EX2001, ¶¶447 and 459.

A POSITA would have known that Cabrera does not disclose “temporally proximate to the operation being performed on the operating file and responsive to detecting the instruction,” at least because the events that initiate the backup / archive occur after the supposed data to be archived has been written and Cabrera does not stipulate any temporal restrictions on when these events in relation to the supposed writing of the data. EX2001, ¶¶445-447.

Schneider fails to cure these deficiencies. The Petitioner states:

As discussed *supra* Section IX.D and IX.E.1.b, it would have been obvious to initiate the backup archive method of Cabrera responsive to interception of a request by an application or operating system to write to disk, as taught by Schneider, as the triggering event to initiate the backup-archive method of Cabrera. EX1003, ¶162.

Pet., 68.

In direct contrast to this statement, a POSITA would have known that Schneider does not disclose “detecting an instruction by a resident program in a computer device for performing an operation on an operating file” for reasons discussed above in Section V.B.2, but instead teaches any sector-based interception of Schneider is uncoupled or independent from an operating system as it teaches a sector-based write. EX1020, 10:28-11:4; EX2001, ¶¶335-340 and ¶¶390-391.

In summary, Petitioner’s Petition fails to show that Cabrera, alone or in combination with Cabrera-Schneider-Whiting disclose “creating an archive file from the operating file and storing the archive file in a temporary storage location temporally proximate to the operation being performed on the operating file and responsive to detecting the instruction.”

Whiting is only relied upon for teaching a “reason for backup is that users may inadvertently delete or overwrite important data files” or administrator duties, e.g., adding new users. Whiting, 1:23-29; 35:3-8. These teachings of Whiting do not cure the deficiencies of either Cabrera or Schneider. For at least these reasons, a POSITA would understand that Cabrera-Schneider-Whiting either alone or in combination do not teach or suggest “creating an archive file from the operating file and storing the archive file in a temporary first storage location temporally proximate to the operation being performed on the operating file and responsive to detecting the instruction.”

4. Petitioner failed to meet its burden in establishing, “identifying presence of the archive file in the temporary storage location responsive to the occurrence of a first event” claim elements [1.4], [8.4], [12.4], [13.4], [14.4], and [15.4]..

Independent claims 1, 8, 9, 10 and 11 recite, in pertinent part, “identifying presence of the archive file in the temporary storage location responsive to the occurrence of a first event.” ‘348 Patent, 8:55-57; (Claim 1, Element [1.4], Claim 8, Element [8.4], Claim 12, Element 12.4], Claim 13, Element [13.4], Claim 14, Element [14.4], and Claim 15, Element [15.5]). Petitioner failed to satisfy its burden for establishing each of these features. Again, Patent Owner notes, Petition has failed to identify with specificity which reference is the primary reference, but rather appears to assert all references are the primary by stating, “Cabrera-Schneider-Whiting.”

The Petitioner asserts:

Cabrera teaches monitoring the “amount of data in the staging area ... in order to identify when a backup or archive session should be initiated.” Pet. EX1005, 3:45-47. Cabrera further teaches an “event monitor 152” that “may be implemented in a wide variety of ways” in order “to watch for desired triggering events and to initiate appropriate action when the events occur.” EX1005, 13:1-10. Cabrera also teaches determining “whether more data should be appended to the staging file,” where “[t]his

determination may be made based on any triggering event.” Id., 11:22-27.

Pet., 69-70.

As shown from the above Petitioner’s mapping of Cabrera does not address several of the claim features including but not limited to

“identifying presence of the archive file,” “archive file” or “responsive to the occurrence of a first event.” Accordingly, the Petition improperly shifts the burden of deciphering Petitioner’s arguments onto Patent Owner and the Board, which, as we discussed above, is contrary to the statutory and regulatory provisions governing *inter partes* review proceedings, as well as the best practices outlined in the Office Patent Trial Practice Guide. *Whole Space Industries Ltd. v. Zipshade Industrial (B.V.I.) Corp.*, IPR2015-00488 Paper 14 at 18 (July 24, 2015) (citing 35 U.S.C. § 312(a)(3); 37 C.F.R. §§ 42.22(a)(2), 42.104(b)(4)–(5)).

Moreover, Patent Owner submits that Cabrera discloses,

The amount of data in the staging area may be monitored.

EX1005, 3:45-47.

This teaching in Cabrera indicates that one criteria for when a backup or archive session should be initiated is the amount of data in the staging area. Determining whether the amount of data in the staging area is at predetermined threshold is not equivalent to “identifying presence of the archive file in the temporary storage

location responsive to the occurrence of a first event.” Also, Cabrera is silent on when the amount of data in the staging area obtained, thus is not responsive to any event and certainly not responsive to a first event. Petitioner has also not identified what the “archive file” or the “first event” is in Cabrera. In addition, Cabrera is silent on when such monitoring occurs and by no means says the monitoring is responsive to a first event. EX2001, ¶¶460-470.

In addition, the Petitioner further appears to rely on another excerpt of Cabrera at 11:22-27; Cabrera further discloses:

In yet another embodiment, the triggering event may comprise the presence of a certain amount of data in the staging file. In the embodiment illustrated in FIG. 5, the triggering event is the expiration of a time delay. Thus, decision block 130 determines whether it is time to establish a remote storage connection. If not, then execution proceeds to decision block 138 where a determination is made as to whether more data should be appended to the staging file. This determination may be made based on any triggering event, as previously explained in conjunction with decision block 124 and decision block 130. If more data that should be appended to the staging file exists, then the data is appended in step 140. In either case, execution proceeds back to decision block 130 in order to wait for the occurrence of the second

triggering event that will initiate connection to remote storage.

EX1003, 11:16-31.

As set forth above, Cabrera does not teach or suggest, to “identifying presence of the archive file in the temporary storage location responsive to the occurrence of a first event.” But rather discloses, “a determination is made as to whether more data should be appended to the staging file ... made based on any triggering event, as previously explained in conjunction with decision block 124 and decision block 130.” Cabrera, 11:23-27. A POSITA would understand is not equivalent to “identifying presence of the archive file in the temporary storage location responsive to the occurrence of a first event.” EX2001, ¶465.

Petitioner does not reference Whiting or Schneider in this section or for teaching “identifying presence of the archive file in the temporary storage location response to the occurrence of a first event.” Accordingly, A POSITA would understand Cabrera-Schneider-Whiting either alone or in combination do not teach “identifying presence of the archive file in the temporary storage location response to the occurrence of a first event.”

Petitioner fails to carry its burden on dependent claims 2-7, 9-11 and 16-17 for the same reasons it failed to carry its burden for underlying independent claims 1, 8, and 12-15. *See, e.g., CallCopy, Inc. v. Verint Americas, Inc.*, IPR2013-00492, Paper 14 at 7 (Feb. 5, 2014) (“Because the grounds asserted against the dependent

claims suffer from the defects of the Case No. IPR2015-01697 Patent 7,525,484 grounds asserted against the independent claims, [the Board] need[ed] to address only the grounds asserted against independent claims.”)

5. No Motivation to combine Cabrera-Schneider-Whiting in the claimed manner.

Petitioner’s Ground 2 challenge also fails because it has not shown that a POSITA would have combined Cabrera-Schneider-Whiting in a manner that would render obvious the Challenged Claims. Petitioner does not assert that either Cabrera, Schneider, or Whiting render obvious the Challenged Claims by themselves.

Patent owner submits that a POSITA would **not** have combined Cabrera-Schneider-Whiting in the claimed manner, at least because (1) Petitioner’s own references teach away from the proposed combination, (2) a POSITA would not have had a reasonable expectation of success at combining the references in the proposed manner, and (3) Petitioner’s purported motivations to combine the references are deficient.

i. Teaching Away

A POSITA would not have a combined Cabrera-Schneider-Whiting, at least because Schneider teaches away from such a combination. “A reference may be said to teach away when a person of ordinary skill, upon reading the reference, would be discouraged from following the path set out in the reference, or would be led in a direction divergent from the path that was taken by the applicant.” *Depuy Spine, Inc.*

v. Medtronic Sofamor Danek, Inc., 567 F.3d 1314, 1327 (Fed. Cir. 2009) (internal quotes and citations omitted). Schneider discourages attempts to use file system level techniques over sector level techniques, and therefore Petitioner's proposed combination (relied on for all Claims in Ground 2) fails.

Schneider discloses:

Keeping historical copies of files as they progress in time has the drawback of eventually forcing the user to manage the archives and purge copies in order to avoid overflowing the disk. Obviously, **one cannot keep a backup of all files on a disk whenever they are changed** for all of time without requiring an unlimited disk, which does not exist.

EX1020, 6:18-22.

A POSITA would have been discouraged from relying on any teachings of Schneider directed towards keeping backups of files as they change or modifying Cabrera-Schneider-Whiting including any file level implementation with any sector-level implementation as Schneider discourages a POSITA from keeping backup copies of files while the files are being changed away from keeping backup copies of files while the files are being changed. *Id.*; EX2001, ¶¶309-310.

Schneider teaches away from combining a file level implementation with any sector-level implementation. Schneider discloses:

Although the present invention at the file level is possible it may be **more difficult to implement, prove correct, and isolate as a subsystem.**

EX1020, 24:20-22.

A POSITA reading the foregoing would have been discouraged from any combination of sector-level implementations of Schneder with Cabrera-Schneider-Whiting at a file level as Schneider teaches it would be more difficult to implement, prove correct and isolate a subsystem. *Id.*, EX2001, ¶¶ 311-312.

Schneider further discloses:

An underlying assumption in the **sector (versus file) based implementation of the present invention** is that the disk will often be left in stable states, thus providing points in time to which one can revert and recover information.

Reading the disk sequentially from start to end is much faster than jumping around on the disk reading each file one at a time. This is because often a file is not stored continuously in one area of the disk, but may be spread out and intermixed with other files across the entire disk.

EX1020, 19:9-11; 2:25-28.

From the foregoing, Schneider teaches that with sector-level backup, disk drives can be read sequentially from start to end, which is faster than jumping around

for each file as files are likely fragmented, spread out and intermixed with other files throughout a disk. A POSITA reading Schneider would understand the file-level implementations and sector-level implementations are fundamentally different, not complimentary, and mutually exclusive alternatives. EX2001, ¶317. A POSITA would have also understood that attempting to apply techniques from one to the other results in undesirable outcomes. *Id.* Additionally, for at least these reasons, a POSITA would have been discouraged from combining any file-level implementation and a sector-level implementation and not led in such a direction. *Id.*

Schneider also teaches:

An underlying assumption in the **sector (versus file) based implementation of the present invention** is that the disk will often be left in stable states, thus providing points in time to which one can revert and recover information.

EX1020, 19:9-11.

Schneider discloses that sector-level archival protects data belonging to files as well as data not belonging to files and provides fault tolerance protection and enhanced resilience that is unavailable with file-level archival via mapping and being coupled to an OS. Schneider discloses:

By implementing the **disk sector embodiment** of the present invention below the operating system (as a pre-disk controller) the present invention is de-coupled from the operating system. Since this embodiment of the present invention can revert a disk back to an earlier state it **can recover from bugs in the operating system** that might otherwise cause catastrophic information loss by a single improper disk write. Backup techniques that are tightly coupled with an operating system and its filing system are less able to recover from bugs in themselves.

EX1020, 10:28-11:4.

There are **two down sides to having the mapping as part of the operating system** proper. First, in the event of an **unexpected crash** the maps may not get fully written out and thus the disk could be left in an extremely confused state. ... The second reason why it is not desirable to incorporate the saving of original states in the operating system is **complexity and bugs**. Intertwining an operating system's filing system with the old state mapping **leads to bugs that can corrupt everything**. However, by separating out the management of the original old disk states from the operating system, this relatively simple management system could recover from the operating system's filing system bugs.”

Id., 13:14-17 and 14:4-9.

From these teachings, A POSITA would have understood that Schneider recognizes that file-level implementations and sector-level implementations are fundamentally different, not complimentary, and mutually exclusive alternatives. EX2001, ¶319. A POSITA would have been discouraged from combining any file-level implementation and a sector-level implementation and not led in such a direction. *Id.*

Schneider further teaches:

By **implementing the disk sector embodiment** of the present invention below the operating system (as a pre-disk controller) the present invention is **de-coupled from the operating system**. Since this embodiment of the present invention **can revert a disk back to an earlier state it can recover from bugs in the operating system that might otherwise cause catastrophic information loss** by a single improper disk write. **Backup techniques that are tightly coupled with an operating system and its filing system are less able to recover from bugs in themselves**.

EX1020, 10:28-11:4.

Accordingly, the sector-level archival is operating system independent or decoupled from an operating system in teaching while file level implementations are tightly coupled with an operating system that approach can lead to catastrophic data loss. A POSITA would have been discouraged from relying on teachings of

Schnieder and understood that attempting to apply techniques from one to the other results in undesirable outcomes. EX2001, ¶¶320-323.

For at least these reasons, Petitioner’s cited references teach away from the proposed combination Cabrera-Schneider-Whiting so strongly that the combination necessarily fails. EX2001, ¶¶304-334.

ii. Reasonable Expectation of Success

Additionally, a POSITA would not have even had a reasonable expectation of success in combining Cabrera-Schneider-Whiting, because a POSITA at the time would have readily understood that teachings specific to a disk/sector level implementation could not be reasonably expected to succeed in a file system level implementation (and vice versa) for similar reasons as discussed herein in Section V.A.iii, *supra*, and the following reasons.

Schneider confirms the difficulty and unlikelihood of success in making the proposed combination. Schneider is primarily directed towards “reverting a disk drive to an earlier point in time.” EX1020, Abs. Schneider also addresses separate strategies for a “file level implementation,” noting that such implementations would “significantly differ” from disk sector approaches and would need to be implemented at an entirely different level (“the operating system’s filing system” rather than “lower down in the disk I/O path”). Schneider does not propose any solutions to combine the two approaches, because Schneider itself recognizes how

difficult (and unlikely to succeed) such modifications would be. For example, Schneider notes that disk/sector based systems rely on fundamentally different assumptions than file based systems. EX1020 at 19:9-10. Schneider points out that even the alternate file embodiment that it suggests would be “difficult to implement, prove correct, and isolate as a subsystem.” EX1020, 24:20-21. Thus again, Petitioner’s submitted references would have suggested to a POSITA that there was no reasonable expectation of success for combining a file system level implementation with a disk/sector level implementation. EX2001, ¶¶311-312, 317, 319, 323, 325, and 327.

iii. Petitioner’s Deficient Motivations to Combine

Furthermore, Petitioner’s asserted motivations to combine the reference are deficient. The deficiencies in the proposed combination are underscored by the fact that Petitioner never even identified which of Cabrera-Schneider-Whiting was the primary reference and which was the secondary references, and instead shifts the burden of deciphering the primary reference from the secondary references onto Patent Owner and the Board, which, as we discussed above, is contrary to the statutory and regulatory provisions governing *inter partes* review proceedings, as well as the best practices outlined in the Office Patent Trial Practice Guide. *Whole Space Industries Ltd. v. Zipshade Industrial (B.V.I.) Corp.*, IPR2015-00488 Paper

14 at 18 (July 24, 2015) (citing 35 U.S.C. § 312(a)(3); 37 C.F.R. §§ 42.22(a)(2), 42.104(b)(4)–(5)).

In light of these defects, the Petition does not provide plausible specific modifications as required to meet its burden, but instead at best relies on generic statements such as (1) “A POSITA would have been motivated to use detecting or intercepting ‘a request to write’ ... as taught by Schneider to ... initiate Cabrera’s backup-archival process,” and (2) “maintain a complete history of the changes made to ... a particular file.” (these statements may indicated that Cabrera should be modified as the primary reference, but it is unclear and shifts the burden to the Patent Owner and Board to decipher). Pet., 67. As discussed above, a POSITA would have been discouraged from attempting any combination of the teachings of Schneider with Cabrera-Schneider-Whiting at least because the references at the time of invention the references teach away (e.g., by decreasing the reliability of the backup including having a potential for a catastrophic loss of data) from such a combination and a POSITA would not have had a reasonable expectation of success in making any of the proposed combinations for reasons discussed herein. EX2001, ¶¶308-334.

For at least these reasons, Petition has not explained why one of ordinary skill in the art would be motivated to combine or modify Cabrera-Schneider-Whiting and what would have provided the basis of a reasonable expectation of success. As the Supreme Court noted in *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 418 (2007),

there must be “a reason” for a skilled artisan “to combine the elements in the way the claimed new invention does”.

VI. Conclusion

Patent Owner respectfully requests that the Board refuse to institute *inter partes* review for the reasons stated herein.

Dated: February 29, 2024

Respectfully submitted,

By: /Rowan P. Smith/
Rowan P. Smith (Reg. No. 64,198)
Attorney for Patent Owner
Datanet LLC

CERTIFICATE OF SERVICE

The undersigned certifies that pursuant to 37 C.F.R. § 42.6(e), a copy of the foregoing was served via email to counsel of record for Petitioners as follows:

Andrew M. Mason, Reg. No. 64,034 andrew.mason@klarquist.com	Cameron D. Clawson, Reg. No. 73,509 cameron.clawson@klarquist.com
	Sarah E. Jelsema, Reg. No. No. 70,804 sarah.jelsema@klarquist.com
	Frank Morton-Park, Reg. No. 80,750 frank.morton-park@klarquist.com

Dated: February 29, 2024

Respectfully submitted,
By: / Rowan P. Smith /
Rowan P. Smith (Reg. No. 64,198)
Attorney for Patent Owner
Datnet LLC

CERTIFICATE OF COMPLIANCE

Pursuant to 37 C.F.R. § 42.24(d), the undersigned hereby certifies that this brief complies with the type-volume limitation of 37 C.F.R. § 42.24 because this brief contains 13,812 words as calculated by Microsoft Word, excluding the items exempted under 37 C.F.R. § 42.24.

Dated: February 29, 2024

Respectfully submitted,

By: /Rowan P. Smith /
Rowan P. Smith (Reg. No. 64,198)
Attorney for Patent Owner
Datanet LLC