

Dropbox, Inc. and Microsoft Corporation v. Datamet, LLC

IPR2024-00078 | U.S. Patent No. 8,473,478

IPR2024-00079 | U.S. Patent No. 9,218,348

Demonstratives of Petitioners Dropbox and Microsoft
February 13, 2025

Klarquist

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Claim 1

'478 Patent

[1.1]	In a computing device, a method for archiving files comprising:
[1.2]	detecting an instruction by an operating system to perform an operation on an operating file;
[1.3]	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;
[1.4]	Searching the first temporary storage location for the archive file responsive to the occurrence of a first event; and
[1.5]	moving the archive file to a second storage location responsive to a second event, the second storage location being a permanent storage location;
[1.6]	after storing the archive file in the first temporary storage location, updating a database to indicate that the archive file is located in the first temporary storage location;
[1.7]	determining a final destination for the archive file;
[1.8]	moving the archive file from the first temporary storage location to an intermediate storage location;
[1.9]	updating the database to indicate the archive file is located in the intermediate storage location; and
[1.10]	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.

'478 Petition (Paper 2), 12 (enumerating EX1001 ('478 patent), Claim 1)

'348 Patent

[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 Petition (Paper 2), 6 (enumerating EX1001 ('348 patent), Claim 1)

Ground 1 Issues in Dispute

- Obviousness of combining Kenley and Long *See* '348 POR (Paper 17), 28-31; '478 POR (Paper 18), 28-32
- Element 1.4 (and its proper construction)
 - '348 [1.4]: “identifying presence of the archive file in the temporary storage location ...”
 - '478 [1.4]: “searching the first temporary storage location for the archive file ...”
See '348 POR, 8-16; '478 POR, 5-17
- Elements 1.6, 1.9, 1.10
 - [1.6]: “... updating a database to indicate that the archive file is located in the first temporary storage location”
 - [1.9]: “updating the database to indicate that the archive file is located in the intermediate storage location”
 - [1.10]: “... updating the database to indicate that the archive file is located in the second storage location”
See '348 POR, 17-25; '478 POR, 17-28

Ground 1 Combination: Kenley-Long

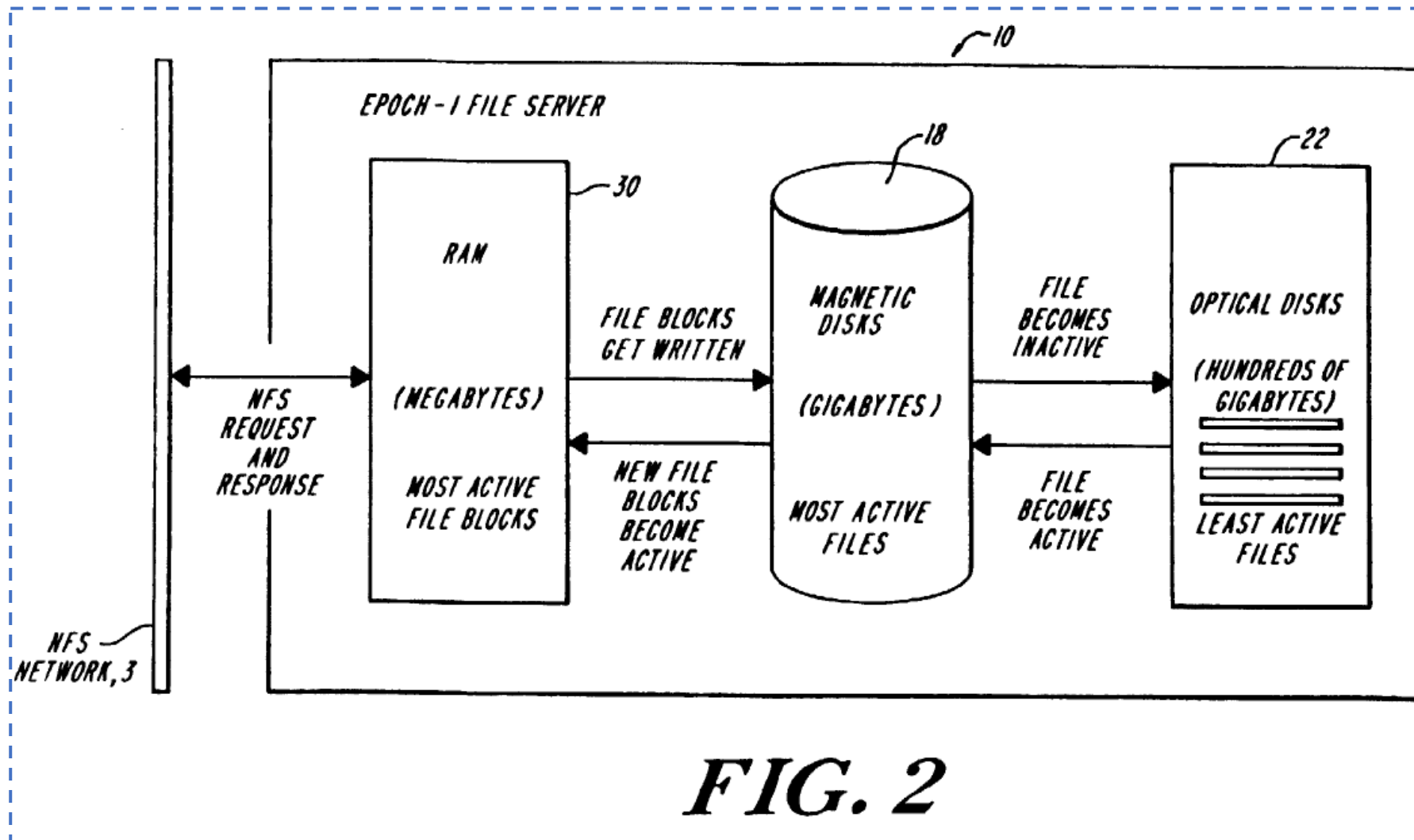
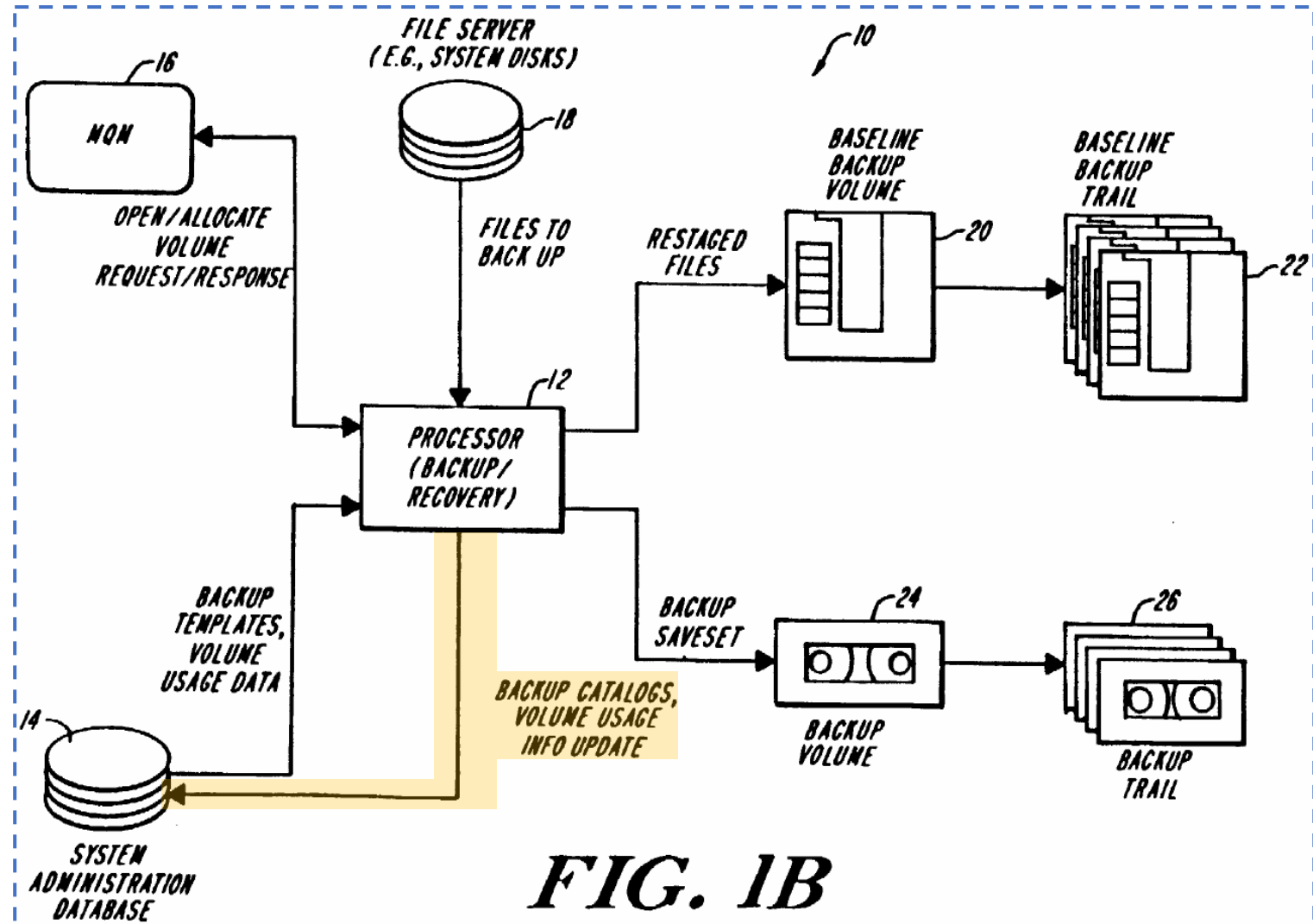


FIG. 2

EX1014 (Kenley), FIG. 2
(cited by '348 Petition
(Paper 2), 56)

“The system further includes a **system catalog** element, in electrical communication with at least the second storage element, for storing and providing transparent access to **catalogued file attribute data** representative of the attributes of the data files stored in the secondary storage element, and for **maintaining the catalogued file attribute data** after the data contained in the selected data files have been **migrated to backing store**.”

EX1014, 3:17-32 (cited by '348 Petition (Paper 2), 8, 14-16, 23-24, 29-30; '478 Petition (Paper 2), 14, 20-22, 35-36) (highlighting added)



EX1014, FIG. 1B (cited by '348 DI (Paper 8), 12; '478 DI (Paper 9), 12) (highlighting added)
DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

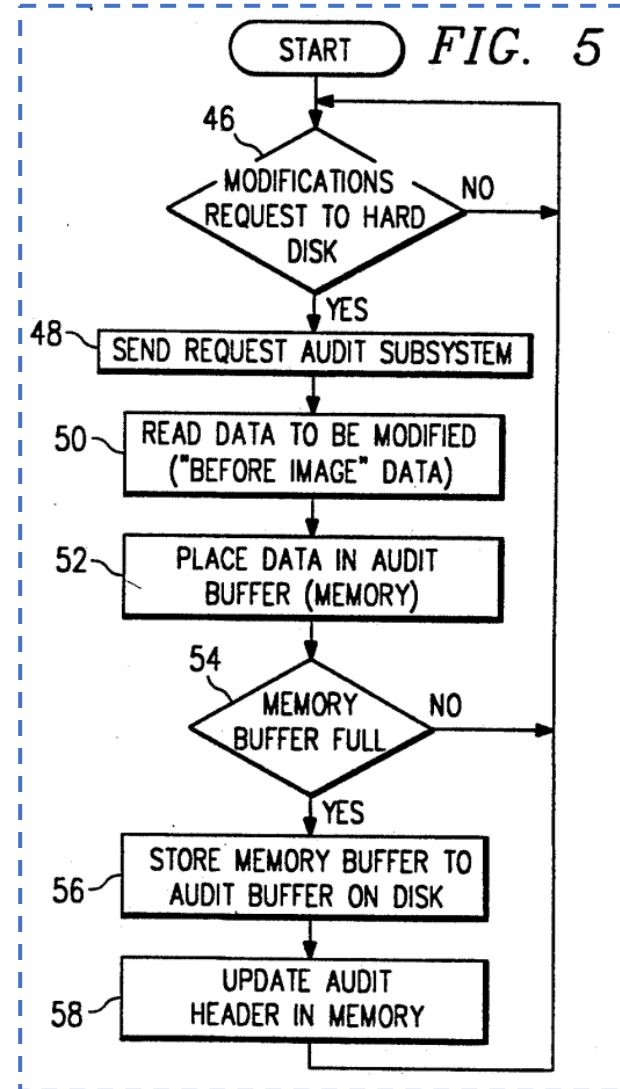
Long

25 In particular, the data on the hard disk is frequently changed during operation of an application program or by the operating system. A system failure may result in valuable information being lost or corrupted.

EX1004 (Long), 4:23-27 (cited by '348 Petition (Paper 2), 13-14, 20, 53-54; '478 Petition (Paper 2), 19-20, 26-27) (highlighting added)

A "modification request" is a request from an application to modify the contents of the hard disk. FIG. 5 is a flow chart describing the sequence of operations performed by the audit subsystem in response to a modification request. 30

EX1004, 7:28-32 (cited by '348 Petition, 9, 14, 18-19, 22, 53-54; '478 Petition, 15, 20, 25-26, 29) (highlighting added)



EX1004, FIG. 5
(cited by '348
Petition, 10, 19, 21-
23, 34-35, 37, 52;
'478 Petition, 16-17,
26, 28-30, 41, 43, 57)

Ground 1 Combination: Kenley and Long

“So implemented, the resulting Kenley-Long system would, based on computer-executable code within the system:

- Generate a modification request, as taught by Long, any time the operating system or a user application ‘need[ed] to modify the contents of’ a file stored on the hard disk. For example, a text-editing program issuing a write operation for a file would be received by the computer system and then ‘sent to the audit subsystem.’ EX1004, 7:27-53.
- Carry out the audit and archive technique taught by Long, on the file being modified, including the copying of files first to the memory buffer, second to the audit buffer portion of the hard drive, and then to external storage, as taught by Long, thus creating ‘archive copies’ of files as taught by Kenley. EX1014, Abstract.
- Maintain relevant audit header information, as taught by Long, in a backup catalog in a relational database, as taught by Kenley. *Supra* ¶55.”

’348 EX1003, ¶58 (cited by
’348 Petition (Paper 2), 18)
(highlighting added);
’478 EX1003, ¶58 (cited by
’478 Petition (Paper 2), 24)

Rationales for combining the teachings of Kenley and Long

1. Kenley-Long improves completeness of file backups
2. Long's technique improves Kenley's similar system
3. Long's technique is applicable to files
4. Kenley and Long's backup storage architectures are complementary
5. Kenley's system catalog dovetails with Long's audit header teachings

'348 Petition (Paper 2), 13-18

'478 Petition (Paper 2), 19-25

Kenley-Long Combination Improves Upon Kenley

EX2020, ¶150. As I explained in my First Declaration, the proposed combination implements Kenley’s file-based backup system with the modification-driven archival approach taught by Long, which achieves “increased reliability by creating backup data after each modification, thus avoiding the potential loss of data created in between the timed backups of Kenley.” EX1003, ¶¶49-51. The proposed combination does not destroy the principle of operation of Kenley in a way that renders the combination not obvious, but rather improves Kenley “by adding Long’s event-driven backup process that efficiently saved backups prior to any modifications of a file stored on hard disk.” EX1003, ¶52.

’348 EX1024, ¶27 (cited by
’348 Pet. Reply (Paper 20), 9)
(highlighting added);
’478 EX1024, ¶32 (cited by
’478 Pet. Reply (Paper 21), 12)

Long's Technique Improves Kenley's Similar System

52. First, Long teaches a “Fault Tolerant Computer With Archival Rollback Capabilities” that “includes significant improvements in hard disk error recovery” and “provides a[n] efficient, reliable and cost[-]effective architecture for disk fault tolerance.” EX1004, 1:2-3, 1:62-64, 2:13-14. A POSITA would have thus been motivated to improve Kenley's file-based “hierarchical digital data storage system” (EX1014, 2:33-36, 3:1-47) by adding Long's event-driven backup process that efficiently saved backups prior to any modifications of a file stored on hard disk.

'348 EX1003, ¶52 (cited by '348 Petition (Paper 2), 14) (highlighting added);

'478 EX1003, ¶52 (cited by '478 Petition (Paper 2), 20)

Long's Technique Was Applicable To Files

53. Second, a POSITA would have known that Long is applicable to files.

...

As Kenley explains, “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.

EX1014, 3:29-32, 7:19-62 (explaining how the file-based approach enables the system to “properly backup and restore files”).

'348 EX1003, ¶53 (cited by '348 Petition (Paper 2), 15) (highlighting added);

'478 EX1003, ¶53 (cited by '478 Petition (Paper 2), 21)

Kenley and Long's Backup Storage Architectures Were Complementary

54. Third, the two references are complementary, as each teaches at least three tiers of storage—primary storage (e.g., RAM or semiconductor memory buffer), secondary storage (e.g., hard disk), and archive or backing storage (e.g., removable or external storage devices)—for use in working with and backing up data.

'348 EX1003, ¶54 (cited by '348 Petition (Paper 2), 15) (highlighting added);

'478 EX1003, ¶54 (cited by '478 Petition (Paper 2), 22)

Long's Audit Header Teachings Pair Well With Kenley's System Catalog

55. Fourth, Kenley's teaching of a database-held system catalog that "stor[es] and provid[es] transparent access to catalogued file attribute data" (EX1014, 3:17-24) dovetails with Long's teachings on an audit header for similarly maintaining data on the current location of backup copies of data (e.g., EX1004, 7:47-53). Kenley's system catalog "enhances transparent access to file attribute data, independent of whether the file is stored in the secondary storage element or has been previously migrated from the secondary storage element" to the backing store. EX1014, 3:25-28. 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." *Id.*; see EX1004, 7:47-53, 9:7-9 (describing

'348 EX1003, ¶55 (cited by '348 Petition (Paper 2), 16, 18) (highlighting added);

'478 EX1003, ¶55 (cited by '478 Petition (Paper 2), 23)

Schneider-US Described Applying Long to Files

57. The reasonable expectation of success, and overall obviousness, of implementing Long’s technique for a file-based system such as Kenley’s is confirmed by another prior-art patent, U.S. 6,016,553 (EX1016; “Schneider-US”) that expressly describes use of Long’s audit buffer technique for saving a file’s data before the file is altered. EX1016, 46:36-47:8 (using file overwrite operation to explain “what normally happens when data is written” and then describing how the Long technique (referred to as the “Move” method) saves a file just prior to it being overwritten); *see also id.*, 46:25-26 (discussing “fundamental methods for saving the

³⁴⁸ EX1003, ¶57 (cited by ³⁴⁸ Petition (Paper 2), 18; ³⁴⁸ EX1024, ¶¶11-13) (highlighting added);
⁴⁷⁸ EX1003, ¶57 (cited by ⁴⁷⁸ Petition (Paper 2), 24; ⁴⁷⁸ EX1024, ¶¶16-18))

Patent Owner Arguments Against Ground 1 Combination

POR Arguments Against the Kenley-Long Combination

POR does not address or dispute:

- Benefits motivating the Kenley-Long combination
See '348 Petition (Paper 2), 13; '348 Pet. Reply (Paper 20), 9; '478 Petition (Paper 2), 19; '478 Pet. Reply (Paper 21), 12
- Schneider-US described using Long's technique to back up files
See '348 Petition, 17-18; '348 Pet. Reply, 1-4; '478 Petition, 23-24; '478 Pet. Reply, 4, 6-8

POR instead asserts:

- Supposed “challenges of mixing file system and disk/sector-level techniques”
'348 POR (Paper 17), 28; '478 POR (Paper 18), 30
- Long's discussion of other concepts “teach[es] away” from combination
'348 POR, 29; '478 POR, 31
- “Combining Kenley with Long would require reengineering Kenley, Long, or both references, which would be a significant undertaking”
'348 POR, 30; '478 POR, 31

Kenley-Long: An Obvious Combination, Not Bodily Incorporation

Instead, Patent Owner pre-supposes that Petitioner's case for obviousness is premised on the bodily incorporation of Long's sector-based archiving process into Kenley's file-based archiving process. However, that is not what Petitioner proposes, as summarized in detail above. Moreover, the test for obviousness is not whether the features of one reference (such as Long) may be bodily incorporated into the structure or process of the other reference (such as Kenley), but rather is "what the combined teachings of the references would have suggested to those of ordinary skill in the art." *In re Keller*, 642 F.2d 413, 425 (CCPA 1981). Further, "[a] person of ordinary

'348 DI (Paper 8), 25
(highlighting added);
'478 DI (Paper 9), 25

Applicability Between File- And Sector-Based Techniques

EX2020, ¶¶65-66. But Mr. Jawadi does not explain why the techniques are “rarely applicable.” Mr. Jawadi could not do so, because prior art such as Schneider-US expressly recognized the applicability of Long to file-based systems, as discussed above. As reflected further in other prior art, a POSITA understood the differences between file-based and sector-based approaches to backup, and also knew how to successfully apply concepts from one approach to another without being discouraged. *E.g.*, EX1025, 1:13-5:21 (describing conventional backup techniques

’348 EX1024, ¶15
(cited by ’348 Pet.
Reply (Paper 20), 5)
(highlighting added);
’478 EX1024, ¶22
(cited by ’478 Pet.
Reply (Paper 21), 8)

No Explanation Why File/Sector Differences Would Deter The Combination

Petitioner Microsoft is not to the contrary. Those patent applications establish, at best for Patent Owner, that processing data at the file level (such as in Kenley) involves many different considerations than those involved in processing data at the sector level (as in Long). *See* Ex. 2014 ¶ 33; Ex. 2016 ¶¶ 1–2, 4–5, 13, claim 12; Prelim. Resp. 17–18, 18–19, 24–25. However, neither Patent Owner nor Mr. Jawadi explains in the present record why such differences would have deterred a person of ordinary skill in the art from changing the timing of Kenley’s file archiving process, by triggering the creation of an archive file whenever a file is modified per the teachings of Long. Indeed, one of the Microsoft patent applications cited by Patent

’348 DI, 26 (highlighting added); ’478 DI, 26

Long Does Not Teach Away From Kenley-Long Combination

modification request.” EX1004, 7:30-32. Long’s description of a modification request triggering a backup–archival process does not criticize, discredit, or discourage its use for triggering file backup–archiving in a file-based system.

Neither does Long’s discussion of “error detection and masking” or “error containment.” EX1004, 2:64-68. On the contrary, Long describes these error detection and containment concepts as “one or more” elements of a general “fault tolerant strategy,” and lists them in parallel to recovery and repair. EX1004, 2:64-68; *id.*, 3:1-33 (describing error detection and masking and error containment as aspects of fault-tolerant systems distinct from error recovery).

³⁴⁸ EX1024, ¶23 (cited by ³⁴⁸ Pet. Reply (Paper 20), 4, 7) (highlighting added);
⁴⁷⁸ EX1024, ¶29 (cited by ⁴⁷⁸ Pet. Reply (Paper 21), 7, 11)

Elements 1.6, 1.9, 1.10
(updating a database)

Kenley-Long Uses A Database To Track Backup File Locations (Elements 1.6, 1.9, 1.10)

78. Thus, in view of these complementary teachings, a POSITA would have been motivated with a reasonable expectation of success to implement Kenley-Long such that information on current locations of different backup files, including the relevant audit-header information taught by Long, was stored in the relational database catalog taught by Kenley.

'348 EX1003, ¶78 (cited by '348 Petition (Paper 2), 32) (highlighting added);
'478 EX1003, ¶78 (cited by '478 Petition (Paper 2), 38)

79. And as with other elements, when used to implement a file-based system such as Kenley's, these database updates taught by Long would indicate that the copied file (archive file) is located in the memory buffer (temporary storage location).

'348 EX1003, ¶79 (cited by '348 Petition, 33) (highlighting added);
'478 EX1003, ¶79 (cited by '478 Petition, 39)

Kenley's Database Catalog Enables Access To Files, Regardless Of Where They Are Stored

institution of review. In particular, on this record, we conclude a person of ordinary skill in the art would have been motivated to utilize Kenley's database catalog of archived file storage locations rather than Long's audit header catalog of archived data, because Kenley's database provides transparent access to the cataloged information, independently of where the archived files are stored. *See* Pet. 16, 18, 30, 32, 37–40; Ex. 1003 ¶¶ 55, 58, 76, 78, 90, 93–94.

'348 DI (Paper 8), 29 (highlighting added);
'478 DI (Paper 9), 29

Kenley's Catalog Stored Backup File Locations (Elements 1.6, 1.9, 1.10)

According to a preferred practice, the identifier includes a list of elements, each comprised of three pieces of information: the volume ID, the file starting offset, storage length and the file length. The file ID can refer to either secondary storage or backing storage in the server 18. This enables the backup processor 12 to properly backup and restore files in instances where the storage server 18 copies segments of files to multiple backing stores; or when the storage server 18 marks as

EX1014, 7:51-62 (cited by '348 Petition (Paper 2), 15; '348 Pet. Reply (Paper 20), 16; '478 Petition (Paper 2), 21; '478 Pet. Reply (Paper 21), 17) (highlighting added)

The system further includes a system catalog element, in electrical communication with at least the second storage element, for storing and providing transparent access to catalogued file attribute data representative of the attributes of the data files stored in the secondary storage element, and for maintaining the catalogued file attribute data after the data contained in the selected data files have been migrated to backing store.

EX1014, 3:17-24 (cited by '348 Petition, 8, 14, 16, 30; '348 Pet. Reply, 16; '478 Petition, 14, 20, 22, 36; '478 Pet. Reply, 17) (highlighting added)

backup trail 26 the file characteristics, i.e., name and attributes (including the aforementioned baseline backup location identifier), of all files whose contents have been copied to the baseline backup volumes 20.

EX1014, 8:10-13 (cited by '348 Pet. Reply, 16; '478 Pet. Reply, 17) (highlighting added)

Catalogs Store File Locations (Elements 1.6, 1.9, 1.10)

catalog Generally, a list of available resources. In a computer, a catalog (or directory) is usually considered to be a list containing information such as name, length, type, and **location of files** or of storage space. In a database, a catalog is the data dictionary. *See also* data dictionary.

EX1027, 67 (cited by '348 Pet. Reply (Paper 20), 15;
'478 Pet. Reply (Paper 21), 16) (highlighting added)

Claim Construction: Element 1.4
(’478: “searching” /
’348: “identifying presence of”)

'478: Proper Construction of "Searching ..."

Disputed element: "searching the first temporary storage location for the archive file"

'478 EX1001, 8:49-50 (Element 1.4)

Prosecution History

may be found at pg. 10, ¶42, lines 1-4. The method still further includes searching the first temporary storage location for the archive file responsive to the occurrence of a first event. Exemplary support may be found at pg. 10, ¶43, lines 6-11; pg. 12, ¶45, Fig. 2A element 100. The method still further includes moving the archive file to a

'478 EX1002, 147 (cited by '478 Pet. Reply (Paper 21), 2) (highlighting added)

'478: Proper Construction of "Searching ..."

Disputed element: "searching the first temporary storage location for the archive file"

'478 EX1001, 8:49-50 (Element 1.4)

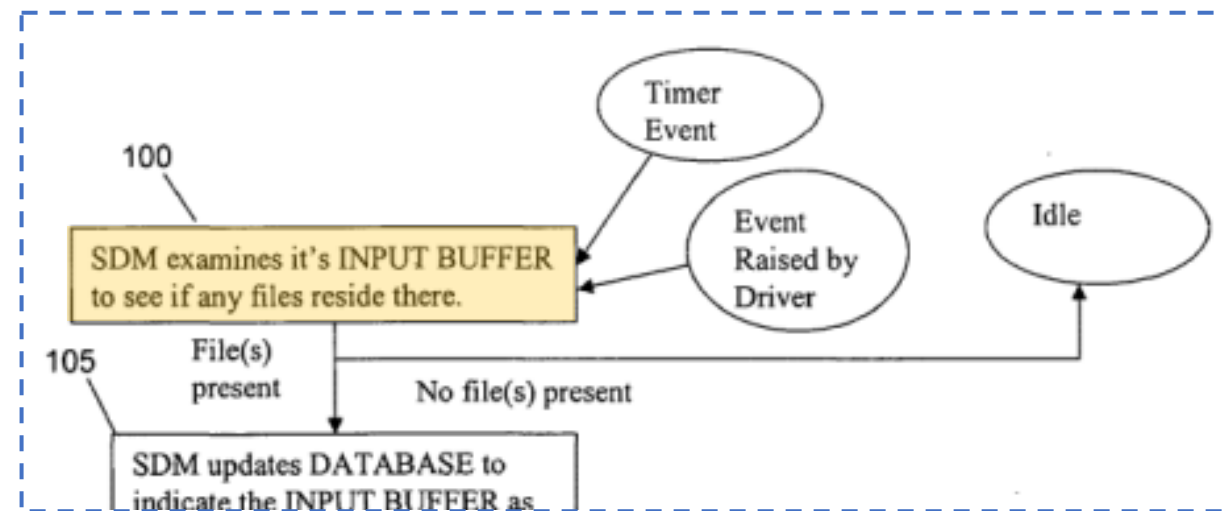
Written Description

15. More particularly smart data management block 15 regularly examines input buffer 20 for the presence of archive files. Smart data management block 15 performs this exami-

EX1001, 5:39-41 (cited by '478 Pet. Reply (Paper 21), 3) (highlighting added)

In step 100 of FIG. 2A, smart data manager 15 examines input buffer 20 to determine whether any archive files are stored therein. If no archive files are present, smart data manager 15 rests idle until the next event occurs. If archive files are detected, in step 105, smart data manager 15 updates database

EX1001, 6:21-25 (cited by '478 Pet. Reply, 3) (highlighting added)



EX1001, FIG. 2A (cited by '478 Pet. Reply, 3) (cropped) (highlighting added)

'478: Improperly Narrow Construction of “Searching ...”

Disputed element: “searching the first temporary storage location for the archive file”

'478 EX1001, 8:49-50 (Element 1.4)

PO construction: “seeking the location of the archive file within the first temporary storage location by comparison of the data therein with a pattern for the archive file”

'478 PO Sur-Reply (Paper 23), 1 (citing '478 POR (Paper 18), 5-7)

Extrinsic Evidence – Dictionary Definitions

search¹ . . . The process of seeking a particular file or specific data. A search is carried out by a program through comparison or calculation to determine whether a match to some pattern exists or whether some other criteria have been met. . . .

search² . . . 1. To look for the location of a file. 2. To seek specific data within a file or data structure. . . .

'478 POR, 7 (quoting EX2023, 424)

'478: Dr. Maier on "Searching ..."

21 Q What does "searching" mean to you?

22 Strike that.

23 How did you use the searching --

24 strike that.

25 In your report, concerning the '478

1 patent, Exhibit 1003 in the 78 case, what

2 definition of searching were you using?

3 MR. MASON: Objection, form.

4 Scope.

5 A (Witness reviewing.)

6 So reading searching here, you

7 know, I discuss this claim element as 1.4,

8 paragraphs 66, and Long.

9 (Witness reviewing.)

10 So using searching in the sense of

11 when we have a collection of information

12 locating -- let's say...

13 Yeah. Let's say identifying an

14 element of that collection.

EX2022, 70:21-71:14 (cited by '478 POR (Paper 18), 7;
'478 PO Sur-Reply (Paper 23), 5) (highlighting added)

'478: Dr. Maier on “Searching ...”

EX1001, FIG. 2A (annotations added). A POSITA, after reading the entire '478 patent, would understand that the plain and ordinary meaning of “searching” as used in the challenged claims to be a general examination or determination of whether any files are located in the first temporary storage location.

'478 EX1024, ¶10 (cited by '478 Pet. Reply (Paper 21), 1, 4)
(highlighting added)

'478: Dr. Maier on “Searching ...”

13. The dictionary-based definitions offered by Mr. Jawadi are narrower than the plain and ordinary meaning of the claim term in view of the intrinsic record, and consequently would require additional steps that are not taught or suggested. For example, if the plain and ordinary meaning of the term “searching” required “seeking a particular file or specific data” as Mr. Jawadi suggests, how would the *particular* file or *specific* data be sought? What is being compared or calculated “to

'478 EX1024, ¶13 (cited by '478 Pet. Reply (Paper 21), 4)
(highlighting added)

'348: "Identifying Presence Of The Archive File"

reading the entire patent. According to the '348 patent, "smart data management block 15 regularly examines input buffer 20 for the presence of archive files." EX1001, 5:44-46. Similarly, "smart data manager 15 examines input buffer 20 to determine whether any archive files are stored therein," and "[i]f no archive files are present, smart data manager 15 rests idle until the next event occurs." EX1001, 6:26-29. Nothing in this description requires any specificity or particularity in "the step of identifying presence of the archive file in the temporary storage location." Instead,

'348 EX1024, ¶30 (cited by '348 Pet. Reply (Paper 20), 11) (highlighting added)

Element 1.4
(’478: “searching” /
’348: “identifying presence of”)

Long Writes Data In The Memory Buffer To Hard Disk; **Kenley** Moves Data In The Form Of Files

Long:

When the memory buffer is full (decision block 54), or the processing unit 14 requires memory used in the memory buffer for another purpose, the audit subsystem **writes the data stored in the memory buffer to the end of the audit buffer 30 on disk (block 56)** and updates the audit header (in the memory buffer) to reflect the changes (block 58).

EX1004, 7:47-53 (cited by '478 Petition (Paper 2), 15-16, 21-22, 24-25, 31, 33, 40; '478 Pet. Reply (Paper 21), 5; '348 Petition (Paper 2), 15-16, 18, 24, 27, 30-31, 34, 36, 51, 53, 57; '348 Pet. Reply (Paper 20), 2) (highlighting added)

Kenley:

A file storage element in the secondary storage **maintains data in the form of files, having attributes** including name, size, owner, protections, time elapsed since a previous access, and type of access—i.e., read or write. In accord with this aspect of the invention, the data selection module includes a data file evaluator, responsive to the data file attributes, for **selecting files to be migrated** from secondary storage to backing store. Ad-

EX1014, 3:1-8 (cited by '478 Petition, 14, 20, 30, 33, 42; '478 Pet. Reply (Paper 21) at 15; '348 Petition, 8, 14, 23, 27, 36; '348 Pet. Reply, 14) (highlighting added)

Kenley-Long Searches For The Archive File To Move It And Track Its Location ('478 Element 1.4)

Furthermore, Mr. Jawadi's arguments focus on implementing Long alone, and fail to consider the obvious implementation of Kenley-Long, where *files* are stored in the memory buffer. Typically, files are selected for migration from a storage location to a different storage location, and as such the system must search the storage location for the files in order to move them. *See, e.g.*, EX1014, 3:5-8 (disclosing "selecting files to be migrated"). Further, Kenley-Long updates a database to indicate the location of an archive file, and so would need to know, regardless of copy method, what files it moved from the memory buffer to the audit buffer on disk in order to appropriately update the database.

'478 EX1024, ¶37 (cited by '478 Pet. Reply (Paper 21), 15) (highlighting added);

'348 EX1024, ¶35 (cited by '348 Pet. Reply (Paper 20), 14)

Kenley-Long Searches The Temporary Storage Location For the Archive File (’478 Element 1.4)

(intermediate storage location). As I explained in my First Declaration, “absent such searching, the system would neither know what it is moving nor when the move has completed.” EX1003, ¶66. Without searching for the archive file, the Kenley-Long system would not know whether there is anything to move to the audit buffer.

’478 EX1024, ¶34 (cited by ’478 Pet. Reply (Paper 21), 13-14) (highlighting added);
’348 EX1024, ¶31 (cited by ’348 Pet. Reply (Paper 20), 11)

Dr. Maier: Kenley-Long Searches For Files, Not Sectors ('478 Element 1.4)

14 Q This argument of citation, is it
15 anywhere in your original declaration?

16 MR. CLAWSON: Objection to form.

EX2025, 66:14-16 (cited by '478 PO Sur-Reply
(Paper 23), 10; '348 PO Sur-Reply (Paper 22), 6)*

17 A (Witness reviewing.)

18 I don't remember whether or not it
19 is cited there.

20 You know, this paragraph is
21 particularly responding to some of
22 Mr. Jawadi's arguments that talk just about
23 implementing Long, and that the point is, if
24 we're using -- using Long with files, then
25 it's not sectors we're looking for in Long to
1 move them, it's files we're looking for in
2 Long to move them.

EX2025, 66:17-67:2 (cited by '478 PO Sur-Reply, 10;
'348 PO Sur-Reply, 6)* (highlighting added)

*PO Sur-Replies erroneously cited EX1025

Kenley-Long Identifies Presence of the Archive File ('348 Element 1.4)

Typically, files are selected for migration from a storage location to a different storage location, and as such their presence in the storage location is identified. *See, e.g.,* EX1014, 3:5-8 (disclosing “selecting files to be migrated”). Further, Kenley-Long updates a database to indicate the location of an archive file, and so would need to know, regardless of copy method, what files it moved from the memory buffer to the audit buffer on disk in order to appropriately update the database.

'348 EX1024, ¶35 (cited by '348 Pet. Reply (Paper 20), 14)

Ground 2 Combination: Cabrera-Schneider-Whiting

POR Arguments Against the Cabrera-Schneider-Whiting Combination

POR does not address or dispute:

- Benefits motivating the Cabrera-Schneider-Whiting combination

See '348 Petition (Paper 2), 63-66; '348 Pet. Reply (Paper 20), 18;
'478 Petition (Paper 2), 63-66; '478 Pet. Reply (Paper 21), 19

POR instead asserts:

- “Cabrera Is Fundamentally Periodic in Nature, and Making the Proposed Change Would Alter the Principle of Operation of Cabrera”

'348 POR (Paper 17), 46; '478 POR (Paper 18), 47

- “Proposed Combination Would Not Work in the System of Cabrera”

'348 POR, 48; '478 POR, 49

Cabrera Teaches Non-Periodic Triggers For Initiating Backup Process

Returning now to FIG. 5, once the triggering event has occurred, whether that be the accumulation of a given amount of data, the expiration of a given time delay, the receipt of a command to stage data, or **any other triggering event**, execution proceeds to step 128 where data is copied from local storage to the staging file. This step may take one

EX1005, 10:56-61 (cited by '348 Petition (Paper 2), 64, 66-68; '348 Pet. Reply (Paper 20), 18-19; '478 Petition (Paper 2), 61, 63, 66-68; '478 Pet. Reply (Paper 21), 19-20) (highlighting added)

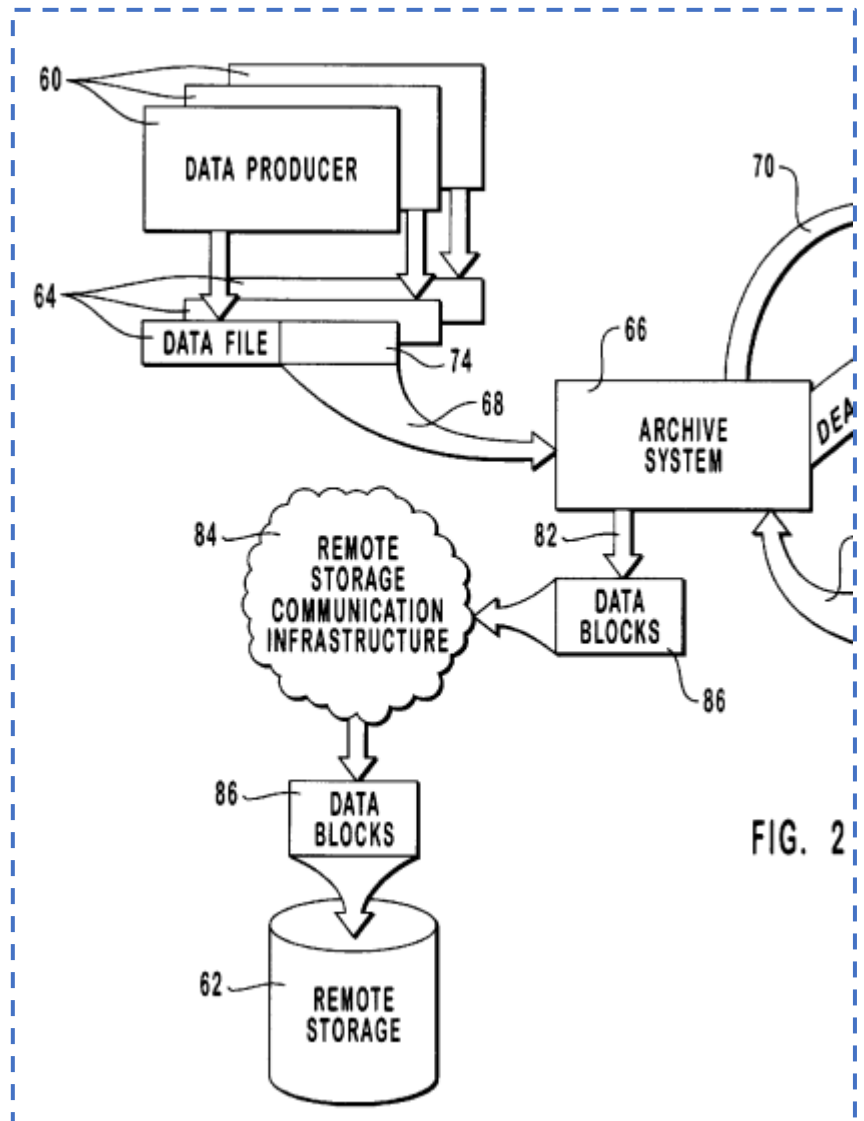
Schneider's Teaching of Copying A File About To Be Modified Would Work With Cabrera

EX1005, 6:63-64). Naturally, the file is already stored on the disk—otherwise, there would be no “file on the disk” to copy “before completing the write to the disk.”

EX1003, ¶159. In view of Schneider's teachings, when a file on the disk is about to be modified or overwritten through a write, the Cabrera-Schneider-Whiting combination creates an archive copy of the file “before completing the write to the disk” and thus overwriting the file on disk. EX1003, ¶¶149, 154-155, 159. Nothing

'348 EX1024, ¶51 (cited by '348 Pet. Reply (Paper 20), 21) (highlighting added);
'478 EX1024, ¶53 (cited by '478 Pet. Reply (Paper 21), 22)

EX1005, FIG. 2 (cited by '348 Petition (Paper 2), 60, 73; '348 Pet. Reply, 21; '478 Petition (Paper 2), 58-59, 74; '478 Pet. Reply, 22-23) (cropped)



Skilled Artisans Had A Reasonable Expectation of Success Implementing Cabrera-Schneider-Whiting

155. The POSITA would have had a reasonable expectation of success implementing the teachings of Cabrera in view of these teachings of Schneider because POSITAs knew how to monitor and intercept instructions to perform an operation on an operating file. For example, Schneider discloses “How to Intercept Disk Activity” so that backup copies can be created prior to writing to a disk. EX1020, 30:4-11. As another example, U.S. 5,684,991 (EX1021, “Malcolm”) discloses intercepting “calls to the operating system to write to files,” which may occur “whenever a program wishes to access a file,” in order to “record modifications to files.” EX1021, 4:2-7; *id.*, 20-25 (“The added software has the

’348 EX1003, ¶155 (cited by ’348 Petition (Paper 2), 65) (highlighting added);
’478 EX1003, ¶134 (cited by ’478 Petition (Paper 2), 64)

Skilled Artisans Had A Reasonable Expectation of Success Implementing Cabrera-Schneider-Whiting

20 Q. Would a POSITA know how to
21 implement that code to detect read or write
22 compiled by other programs?

23 MR. JAFFESS: Object to the
24 form.

25 THE WITNESS: There -- what --

1 we go to the next level. What the
2 POSITA would have to do at that point
3 would be to intercept to write code
4 that intercepts calls. The operating
5 system calls for the file system.

6 For example, the user would --
7 so in those days, the easiest way is
8 there is an interrupt. If a user says
9 open file and then close and whatever,
10 eventually those calls go to the -- go
11 to the DOS -- not DOS. Excuse me --
12 specific interrupt.

13 If a user wants to bypass all
14 of these functions from C and
15 whatever, it can write code and then
16 they can substitute the call, the
17 interrupt handler of interrupt 21,
18 they can reroute it to their own code.

EX1026, 236:20-237:18 (cited by '348 Pet. Reply (Paper 20), 20; '478
Pet. Reply (Paper 21), 21) (highlighting added)

Cabrera-Schneider-Whiting Identifies Presence of / Searches for Archive Files (Element 1.4) When Determining Whether To Add More Files

(a first event). EX1005, 11:24-27. Further, the determination of whether to append more data depends on how much storage space is allocated or allocable for the staging area, which in turn depends on what files are already stored in the staging area. *E.g.*, EX1005, 2:12-13, 2:44-49, 2:55-61. Under the proper construction as discussed *supra* Section V.B, Cabrera's evaluation of the staging area necessarily requires identifying presence of an archive file previously stored in the staging area but not yet migrated; otherwise, the system would not know whether there is allocated or allocable space available to add the additional archive files to the staging area. EX1003, ¶165. This evaluation in Cabrera thus satisfies the element

'348 EX1024, ¶54 (cited by '348 Pet. Reply (Paper 20), 23) (highlighting added);
'478 EX1024, ¶56 (cited by '478 Pet. Reply (Paper 21), 24)

Whiting Discloses A Global Directory Database And Hierarchical Storage (Elements 1.6, 1.9, 1.10)

foData> record **436** for that file. In addition, each entry contains the **<fileID>** record **214** (six bytes) which can be used to locate the actual file data contents. The total size for

EX1006, 21:47-49 (cited by '348 EX1003, ¶156; '348 Petition (Paper 2), 63, 65; '478 EX1003, ¶135; '478 Petition (Paper 2), 65) (highlighting added)

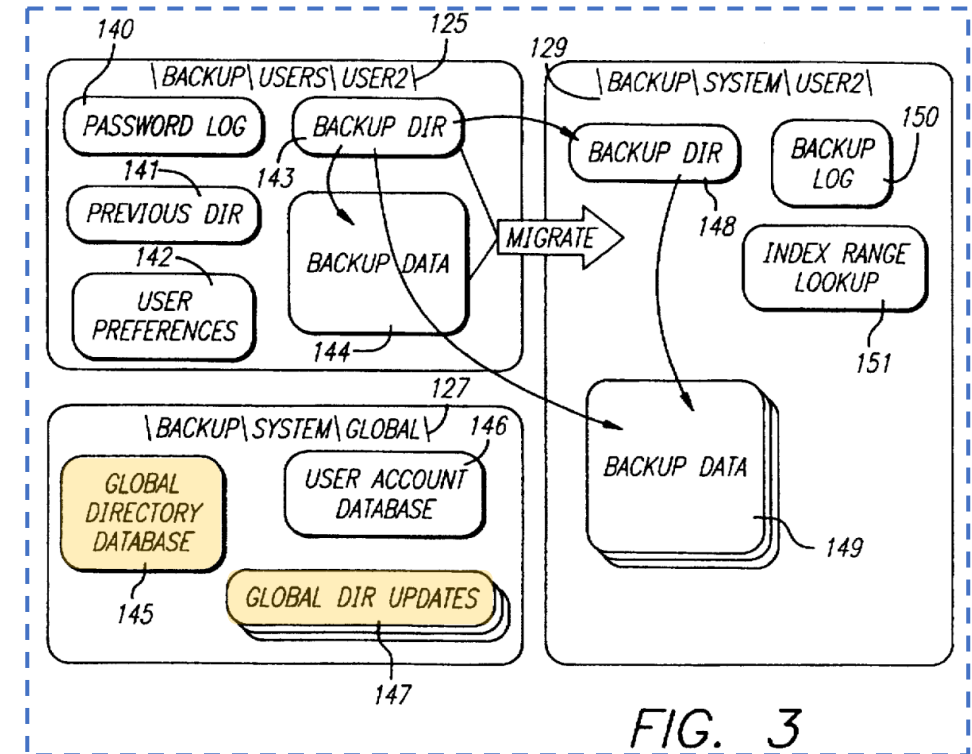
In yet another embodiment, the backup storage means incorporates **hierarchical storage management** (HSM), in which files that have not been accessed for a long time are migrated from disk to a secondary storage means, such as tape or optical disk. The main purpose of HSM is to save on storage costs for very large storage systems by providing the management tools that **allow the migration to be transparent** to the system, except for the additional delay in accessing some files. Use of any form of HSM in conjunction with the

EX1006, 6:52-60 (cited by '348 EX1003, ¶¶153, 156; '348 Petition (Paper 2), 65; '478 EX1003, ¶¶132, 135; '478 Petition (Paper 2), 65) (highlighting added)

Whiting Provides Implementation Details For Tracking Archive File Locations With A Database As Archive Files Migrate

such as Whiting, for relevant teachings. The global directory index database of Whiting enables such transparent access by storing the location of backup data files. EX1006, 10:52-55, 21:47-49 (describing the record in the global database “which can be used to locate the actual file data contents”). The POSITA implementing Cabrera thus would have naturally used these teachings from Whiting to provide implementation details for storing and updating the location of various archived versions of a file, including as the archive files are migrated from local storage to remote storage, so that the archive files can actually be located and accessed when needed by the system or user.

'348 EX1003, ¶156 (cited by '348 Petition (Paper 2), 66) (highlighting added);
'478 EX1003, ¶135 (cited by '478 Petition (Paper 2), 66)



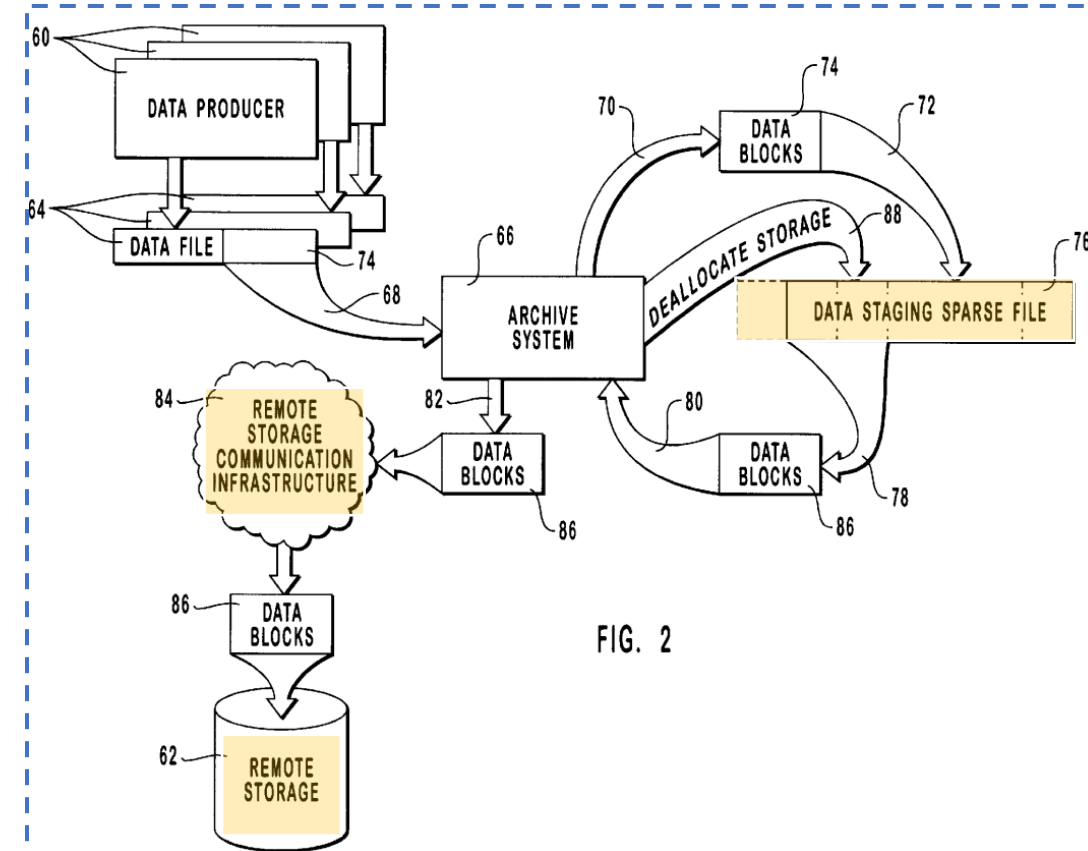
EX1006, FIG. 3 (cited by '348 EX1003, ¶152;
'348 Petition, 63; '478 EX1003, ¶131; '478
Petition, 63) (highlighting added)

Cabrera-Schneider-Whiting Moves The Archive File To An Intermediate Storage Location On Its Way To A Permanent Storage Location

[1.5] When a second pre-determined event occurs, archive system 66 will transfer all or part of the data in sparse file 76 to remote storage 62. Thus, embodiments within the scope of the present invention may comprise means for transferring data from the staging storage area to a remote storage medium. By way of example, and not limitation, in FIG. 2 such means is illustrated by arrows 78, 80, and 82 which illustrate data being moved from sparse file 76 and delivered to remote storage communication infrastructure 84. Such a means may be implemented by any mechanism capable of retrieving data from sparse file 76 and either directly delivering the data to remote storage 62 or to an intermediate system or subsystem which will, in turn, deliver the appropriate data to remote storage.

[1.8]

EX1005, 7:47-60 (cited by '348 Petition (Paper 2), 73; '348 Pet. Reply (Paper 20), 24-25; '478 Petition (Paper 2), 73; '478 Pet. Reply (Paper 21), 25-26) (highlighting added)



EX1005, FIG. 2 (cited by '348 Petition (Paper 2), 60, 73; '348 Pet. Reply (Paper 20), 21; '478 Petition (Paper 2), 58-59, 74; '478 Pet. Reply (Paper 21), 22-23) (highlighting added)

Appendix

Schneider-US Described Applying Long to Files

One approach to retaining discarded data on a more or less continuous basis is described in U.S. Pat. No. 5,325,519, entitled “Fault Tolerant Computer with Archival 0 Rollback Capabilities”, to Long et al. (“’519 patent”). The ’519 patent

EX1016 (Schneider-US), 4:27-30 (cited by ’478 Pet. Reply (Paper 21), 6; ’478 EX1024, ¶17; ’348 Pet. Reply (Paper 20), 3; ’348 EX1024, ¶12) (highlighting added)

The Move Method

The basic elements of the Move Method are described in the ’519 patent. In this method, a portion of the hard disk is

EX1016, 8:7-9 (cited by ’478 Pet. Reply, 6; ’478 EX1024, ¶17; ’348 Pet. Reply, 3; ’348 EX1024, ¶12) (highlighting added)

FIG. 38 illustrates the Move Method. In each frame

...
engine. In Frame 1 the file is overwritten, in RAM at least, but before the hard disk is modified, the affected pages are moved into the history buffer. Frame 1 shows the reading of the data about to be overwritten and where it is eventually placed. However, for the moment the data goes into a buffer.

EX1016, 46:63-47:5 (cited by ’478 Petition (Paper 2), 23; ’478 EX1024, ¶17; ’478 EX1003, ¶57; ’348 Petition (Paper 2), 17-18; ’348 EX1024, ¶12; ’348 EX1003, ¶57) (highlighting added); (see also EX1016, FIG. 38 (cited by ’348 Petition, 17-18; ’478 Petition, 23-24)

Phillips

“Properly viewed, the ‘ordinary meaning’ of a claim term is its meaning to the ordinary artisan after reading the entire patent. Yet heavy reliance on the dictionary divorced from the intrinsic evidence risks transforming the meaning of the claim term to the artisan into the meaning of the term in the abstract, out of its particular context, which is the specification.”

Phillips v. AWH Corp., 415 F.3d 1303, 1321 (Fed. Cir. 2005)

(cited by ’478 Pet. Reply (Paper 21), 1-2; ’348 Pet. Reply (Paper 20), 10)

CERTIFICATE OF SERVICE
IN COMPLIANCE WITH 37 C.F.R. § 42.6(e)(4)

The undersigned certifies that on February 6, 2025, **PETITIONERS'**
DEMONSTRATIVES were served on Patent Owner via electronic mail as
follows:

Ari J. Jaffess
Liston Abramson LLP
405 Lexington Avenue, 46th Floor
New York, NY 10174
Email: ari.jaffess@listonabramson.com

Ronald Abramson
M. Michael Lewis
Gina K. Kim
Liston Abramson LLP
405 Lexington Avenue, 46th Floor
New York, NY 10174
Email: ron.abramson@listonabramson.com
michael.lewis@listonabramson.com
gina.kim@listonabramson.com

Rowan P. Smith
Messner Reeves LLP
7250 N. 16th St., Ste. 410
Phoenix, AZ 85020
Email: rsmith@messner.com

By: /Andrew M. Mason/
Andrew M. Mason (Reg. No. 64,034)
andrew.mason@klarquist.com
KLARQUIST SPARKMAN, LLP
One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, Oregon 97204
Tel: 503-595-5300
Fax: 503-595-5301

Counsel for Petitioners