

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 IPR2024-00080
Patent 10,585,850

**PATENT OWNER'S PRELIMINARY RESPONSE UNDER 37
C.F.R. § 42.107 TO PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 10,585,850**

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

[illegible]

TABLE OF CONTENTS

I.	Introduction	1
II.	Background	1
A.	The ‘850 Patent Background.....	1
B.	The ‘850 Patent Invention	2
C.	The ‘850 Patent’s Novel Claimed Solution	6
D.	Level of Ordinary Skill in the Art	9
E.	Petitioner’s references	9
1.	Schneider [EX1005].....	10
a.	Schneider is focused on media recovery, not file archiving	11
b.	External backup used for crash recovery, not file recovery.....	15
c.	Schneider’s file recovery embodiments	19
2.	Crawford [EX1009].....	20
3.	Sweat [EX1006]	21
III.	Legal Standards	24
IV.	Claim Construction	26
V.	Argument	27
A.	Ground 1: Schneider.....	27
a.	Petitioner’s Claim Mapping is Materially Deficient.....	27
b.	Claims 1, 10, and 18.....	27
i.	The Petitioner has failed to meet its burden in establishing “presenting information for a collection of one or more previous versions of the file ... wherein a restorable representation of each version, V, of the previous versions, is retrievable from a remote storage location...;” claim elements [1.2], [10.2], [18.2]	27
ii.	The Petitioner has failed to meet its burden in establishing “responsive to a selection to preview ... presenting a presentable representation of the selected previous version” claim elements [1.3], [10.3], [18.3]	33
iii.	The Petitioner has failed to meet its burden in establishing “responsive to a selection to restore the selected previous version, retrieving the restorable representation of the selected previous version from the remote storage location and storing the selected previous version ... as the current version on the local storage location” claim elements [1.4], [10.4], [18.4]	35

c.	Claim 19	37
i.	The Petitioner has failed to meet its burden in establishing “(d) previewing selected revision of said file, wherein said selected revision of said file is received from an Internet storage area network;”	37
ii.	The Petitioner has failed to meet its burden in establishing (f) previewing selected another revision of said file, wherein said selected another revision of said file is received from a network attached storage location;”	38
B.	Ground 2: Schneider+Crawford	38
C.	Ground 3: Sweat	42
1.	Sweat is not Analogous Art.....	42
i.	The Petitioner has failed to meet its burden in establishing “responsive to a selection to restore the selected previous version, retrieving the restorable representation of the selected previous version from the remote storage location and storing the selected previous version ... as the current version on the local storage location” claim elements [1.4], [10.4], [18.3]	45
D.	Ground 4: Sweat+Schneider.....	47
VI.	Conclusion	48

I. Introduction

Petitioners Dropbox, Inc. and Microsoft Corporation (collectively “Petitioner”) have challenged Claims 1-21 (the “Challenged Claims”) of U.S. Patent No. 10,585,850 (the “’850 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. Because Petitioner has failed to demonstrate a reasonable likelihood of success of invalidating any of the independent claims, the rest of its challenges necessarily fail. For the following reasons, the Petition should be denied.¹

II. Background

A. The ‘850 Patent Background

The ‘850 Patent is titled “AUTOMATIC REAL-TIME FILE MANAGEMENT METHOD AND APPARATUS.” EX1001. The patent was filed on December 15, 2015, issued on March 10, 2020, and claims priority to Provisional

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

Application No. 60/234,221 (the “’221 Provisional”), filed September 21, 2000.
EX1001 at 1:4-10.

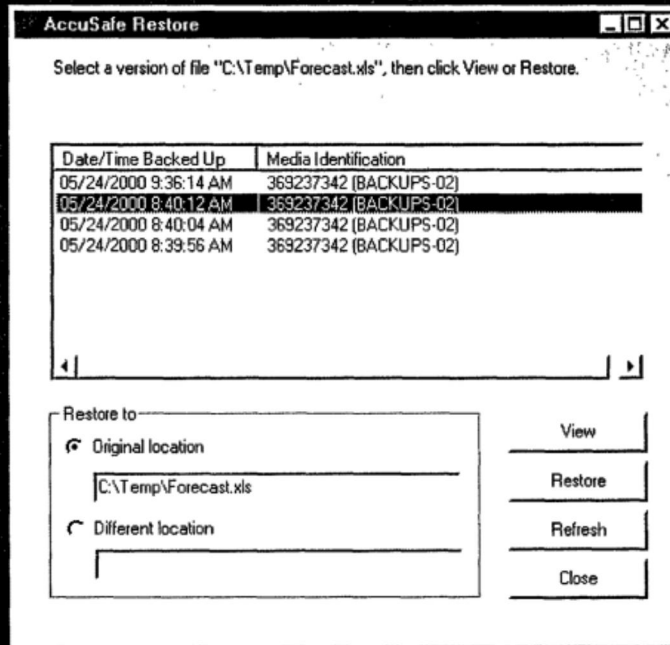
B. The ’850 Patent Invention

The ’850 Patent recites a method of “restoring a file to a previous version of the file.” EX1001, Abstract. The method of restoring a file includes providing the ability for a user to preview a collection of available backed-up versions of a particular file, select one of those versions in order to preview the selected version of the file, and, ultimately, if the previewed version of the selected file is the desired version, restore that version of the selected file. The file that is restored is eventually stored on the user’s computer as the current version of the file, enabling the user to continue working on the file with normal back-up operations continuing.

The following figures and description presents an example use case for the technology of ’850 Patent. The figures are taken from the ’221 Provisional and include screenshots of a prototype Accusafe product configured to enable selective backup file previewing and retrieval.

Using the present invention, for a particular file, information indicative of a list of captured versions of a file are displayed. EX1003, 32.

Example use-case scenario

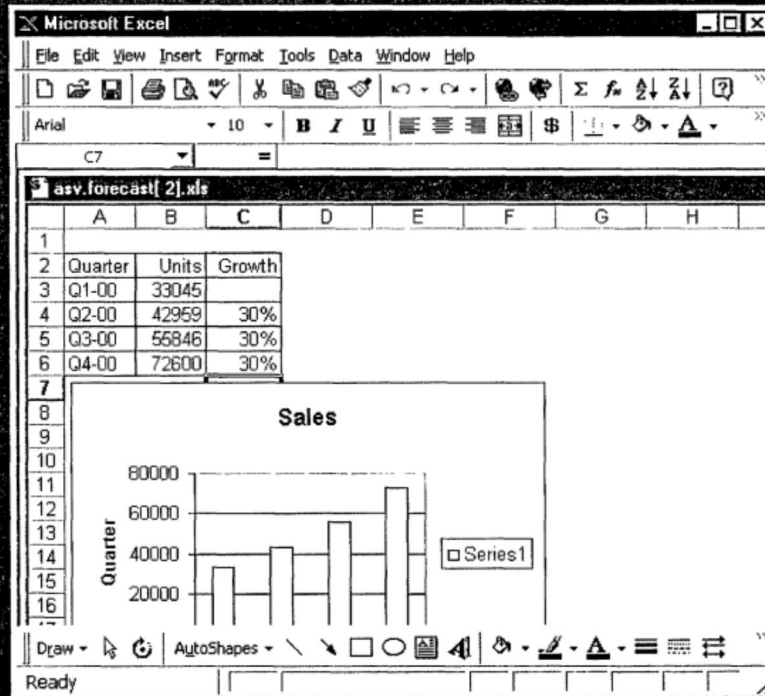


AccuSafe displays the list of revisions it has captured.

The user selects a version from this list, then clicks the View button to determine if this is the version he wishes to restore...

The user can, using the interface provided, select a particular prior version of the file and select the "View" option to preview that particular version of the file before formally restoring the file from the remote storage location. The preview version of the file is depicted below. EX1003, 33.

Example use-case scenario



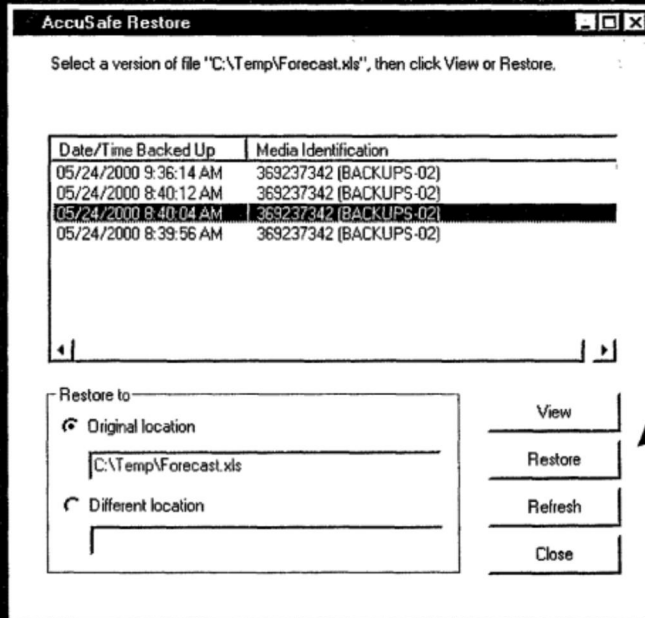
AccuSafe displays the revision selected, using the application that created the file.

If this isn't the version that the user wanted...

If the user wishes to preview a different version of the file, they can close the preview view and select "View" on a different version of the file using the same user interface.

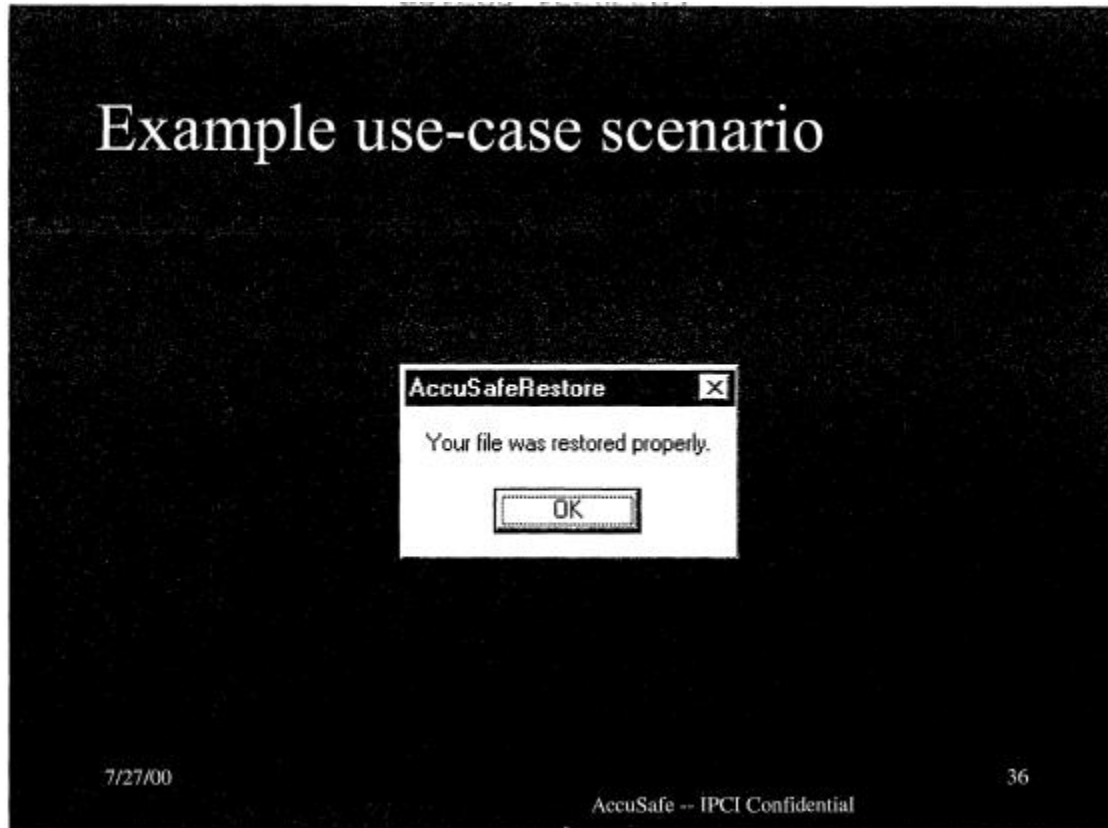
Having found the desired version of the file to restore, the user can select the "Restore" button to formally restore the file. EX1003, 36.

Example use-case scenario



The user has verified this to be the proper revision to restore, so the user clicks Restore to retrieve his file.

The user is notified when file restoration has been completed, enabling the user to resume normal operations on the computer. EX1003, 38.



C. The '850 Patent's Novel Claimed Solution

The '850 patent claims are directed to methods of "restoring a file to a previous version of the file." EX1001, 8:63-64. The claimed methods involve, for a particular file, "presenting information for a collection of one or more previous versions of the file" where the information includes information indicative of at least one or more previous versions of the file. EX1001, 8:66-67. An example of a graphical user interface that may display such information is shown above.

The claims state that, for each previous version of the file, a "restorable representation" of each version of the file is "retrievable from a remote storage

location.” EX1001, 9:3-5. Additionally, each of the restorable representations of each version of the file “ha[s] at least information required for recovering the version V by [the user’s] computational machine.” EX1001, 9:5-7.

Having presented the information for the various versions of the file, the claimed method, “responsive to a selection to preview a selected previous version of the file ..., present[s] a presentable representation of the selected previous version” of the file. EX1001, 9:10-13. In this manner, the claimed invention enables a user to selectively preview different available versions of a backed-up file.

If the previewed file is the file the user wishes to restore, the claimed method includes, “responsive to a selection to restore the selected previous version, retrieving the restorable representation of the selected previous version from the remote storage location.” EX1001, 9:21-24. Once retrieved, the corresponding “selected previous version” of the file is, as part of the restoration process, stored by the user’s computation machine “as the current version.” EX1001, 23-25. The “selected previous version” is available “from the restorable representation” that was retrieved from the remote storage location responsive to the user’s request to restore the “selected previous version.” EX1001, 23-25 21-28.

When restoring a file, the file is stored on the user’s computer as the “current version” of the file. This can, for example, enable the user to resume normal editing of the file with back-up operations being ongoing.

The claim language of Claim 1 is reproduced below for reference:

[1.1]	A method of restoring a file to a previous version of the file, a current version of the file being available at a local storage location, comprising the steps of:
[1.2]	(A) presenting information for a collection of one or more previous versions of the file, the information for the collection including information indicative of at least one or more of previous versions of the file, wherein a restorable representation of each version, V, of the previous versions, is retrievable from a remote storage location, the restorable representation having at least information required for recovering the version V by a computational machine with the current version being accessible, the remote storage location being accessible through a network;
[1.3]	(B) responsive to a selection to preview a selected previous version of the file based on the presented information for the collection of (A), presenting a presentable representation of the selected previous version, the selected previous version being one of the previous versions of the file in the presented information for the collection, the presentable representation having at least information required for presenting at least a portion of the selected previous version by the computation machine with the current version being accessible; and
[1.4]	(C) responsive to a selection to restore the selected previous version, retrieving the restorable representation of the selected previous version from the remote storage location and storing the selected previous version by the computational machine as the current version on the local storage location, the selected previous version available from the restorable representation of the selected previous version.

EX1001, Claim 1 at 8:96-9:28.

D. 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,² 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.

E. Petitioner’s references

Petitioner’s invalidity grounds are based on PCT Publication No. WO 99/12101 (“Schneider”), U.S. Patent No. 7,062,532 (“Sweat”), and U.S. Patent No. 5,771,354 (“Crawford”). Petitioner relies on four separate obviousness grounds:

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

Ground	Claims	References
1	1-21	Schneider
2	1-21	Schneider+Crawford
3	1-8, 10-16, 18- 21	Sweat
4	9, 17	Sweat+Schneider

Pet. at 2.

1. Schneider [EX1005]

Although Schneider and the ‘850 Patent both generally describe data integrity solutions, they are focused on different technological problems and present different (and non-compatible) technological solutions to those problems.

As discussed above, the ‘850 Patent is directed towards improvements to a **file archiving** system. The claimed methods enable a user to easily browse and recover different versions of a particular file. A typical use case for this solution may involve a user who has made changes to a file over several days or weeks browsing through those old versions of the file to recover work that was accidentally deleted or changed some time ago.

In contrast, Schneider presents, primarily, a **media or hard drive recovery** system that enables a computer user or system administrator to recover from a hard

drive or other media failure. Schneider describes a short-term hard drive recovery solution that is designed to enable a user to “roll-back” a hard drive to a state as the hard drive existed before a failure. EX1005, 10:9-16. To enable that roll-back operation, Schneider’s solution is designed to capture changes to the various sectors of a hard drive as the changes are made and insert those changes into a circular buffer. As the computer continues its operation, the circular buffer is updated each time a hard drive sector is changed. Old data is deleted. EX1005, 12:1-10. While this is an effective approach for relatively short-term hard drive recovery, it complicates any attempts to use Schneider’s approach for file recovery. In fact, as made clear below, although Schneider claims to be adaptable to file archiving purposes, Schneider requires an approach to file archiving that is incompatible with and does not render obvious the claims of the ‘850 Patent.

a. Schneider is focused on media recovery, not file archiving

Schneider presents a hard drive recovery solution. Specifically, Schneider presents a “method and apparatus for reverting a disk drive to an earlier point in time[. 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.” EX1005, Abstract. As described by Schneider, the use of a circular buffer enables the storage of **recent changes** to sectors of a hard drive to a buffer having a predetermined maximum size. If a drive failure occurs, data can be retrieved

from the circular buffer to “roll-back” the state of the hard drive to a recent point in time.

Because Schneider is focused on a media recovery solution, Schneider is focused on *short-term* storage of *all recent changes* made to a hard drive. Schneider’s hard drive recovery solution does not provide archiving on a file-by-file basis. As described by Schneider:

“It is generally recognized that most information recovery is to **fairly recent points in time**. Therefore, it is advantageous to manage the storage required for backup **not by file, but by time**. Using the technology of the present [invention], information is maintained for **a reasonable period of time and then is automatically discarded**. What is included in the backup information is significantly all of the activity to the disk. This allows a user to return to any disk state at any time up to a limit. This limit is determined by the amount of available backup storage and the rate to which information is written (user activity).” EX1005, 10:9-16 (emphasis added).

In Schneider’s short-term media recovery solution, as changes are made to a hard drive’s sectors, those changes are intercepted and stored into a circular history buffer. Schneider states, “[t]he 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. As already stated, it may take days before a user's write activity transfers so much information that the history buffer wraps and very old states are discarded.” EX1005, 12:1-10.

In the event of a disk failure, Schneider uses the data stored in the history buffer to enable a user to recover a version of the hard drive as it existed prior to the disk failure. Schneider describes two ways to enable this roll-back operation.

In the first case, a user who was previously running a drive C may wish to access an old version of that drive. In that case, assuming the user's computer system can “support another drive D, which instead of being a real disk drive, is a simulated or virtual drive whose image is created by combining information on the original drive C and the history buffer (which is typically part of drive C). The process of looking back in time involves setting a reference time for [the simulated] drive D and then simply accessing it as if it really was another physical disk drive whose contents had been copied from drive C at the specified time.” EX1005, 15:24-29.

In the second approach, “the main drive C [is reverted] back to an earlier state. In this situation there is no simulated D drive from which old information is brought forward into the present, but simply the main drive is entirely brought back in time.

This recovery mode is particularly useful when the current state has become unusable (cannot boot or access files) or undesirable (an installation of new software or hardware drivers does not work as expected). The implementation can be as simple as copying the appropriate saved original data back into place, updating the history buffer to reflect the reversion, and restarting the system.” EX1005, 16:9-17.

Schneider describes how, to some degree, this technology could enable a user to access old version of a particular file:

“In a practical sense, this means that while in a text editor, one could set the time for drive D to 11:30, open a desired ‘old’ file, decide they want an even earlier version, adjust the time [for drive D] back to 11:15, and try again. It would not matter if at the current time the file had been deleted, or even its directory deleted, since by setting a time for drive D you are referencing back in time for drive C. This approach of simulating a new drive whose backup time can be entered **is far more flexible than dealing with real copies** (like tape) which will only be available at specific times (when the backups were requested).” EX1005, 16:1-8 (emphasis added). In this manner, Schneider’s solution appears to teach away from individual file archiving operations in which multiple versions of specific files (rather than a historic log containing a stream of hard drive sector changes) are maintained to allow users to directly access old versions of those files without first performing a full drive recovery.

Schneider does acknowledge the existence of more convenient file archiving technologies but indicates that those technologies are readily distinguishable from the hard-drive recovery teachings of Schneider. Schneider states, “[s]oftware exists to perform incremental backups to, for example, a tape drive. When viewing the tape, you can specify what version of a file you want to access. However, the technology of the present invention whether logging sector or file transfers provides for eliminating the specific backup step.” EX1005, 18:7-10. Schneider further states, “[k]eep in mind that the present invention is not designed to replace traditional tape backup approaches, as they offer recovery from physical drive failures and do guarantee that information is available from a specific time. Such a backup is totally guaranteed in terms of what it provides. The present invention **on the other hand**, can only go back in time as far as the history buffer's size permits. The amount of time depends on the user's write activity.” EX1005, 18:23-28 (emphasis added).

b. External backup used for crash recovery, not file recovery

Schneider anticipates that for a given installation, the history log, which stores a history of hard drive sectors changes, could become inaccessible in the event of a drive failure because the historic log is stored on the user's computer's hard drive. As described by Schneider:

“The fault tolerance provided by the various methods deals with non-hardware failures like the user accidentally overwriting a file or a bug in an application corrupting files. However, there is also the case of the

disk actually ceasing to function (i.e., if it broke and the information it contained is lost). Recovery from such a failure typically involves installing a new hard disk, re-installing the operating system, and then restoring files from a backup tape or similar device. This is a time-consuming process and often involves some loss of data, that which was affected after the backup.” EX1005, 108:18-26.

To mitigate this problem, Schneider contemplates that the historic log (i.e., the log file storing hard drive sector changes which is stored locally on the user’s computer), along with other data necessary to recover a hard drive could be stored in an external drive or disk. In that case, Schneider presents an “entirely different approach to achieving an external backup [that] is to have an external disk drive that, like the main disk, employs a method of saving original disk states. Thus, instead of creating a backup of a specific point in time, the information on the backup includes the historical information, **allowing the backup to re-create a range of ‘backup’ times**. In other words, the external disk generally mirrors the main internal disk. This is how a RAID system generally works.” EX1005, 109:6-11.

As described by Schneider, the backup data that is stored on the external disk is only used in the case of a hard drive failure. During all other computer operations, if backup data is to be retrieved (and particularly whenever recovering a single file), Schneider relies on its local backup data, which is stored on the user’s computer and is readily accessible.

Schneider describes the use of the external disk as follows:

“All the methods thus far presented for saving original disk states are conceptually designed around a single disk. Of course, more than one disk may be involved, with their collective storage pooled into one large logical disk. The fault tolerance provided by the various methods deals with non-hardware failures like the user accidentally overwriting a file or a bug in an application corrupting files. However, there is also the **case of the disk actually ceasing to function (i.e., if it broke and the information it contained is lost)**. Recovery from such a failure typically involves installing a new hard disk, re-installing the operating system, and then restoring files from a backup tape or similar device.” EX1005, 108:16-24 (emphasis added).

“**Should a crash occur** and the lagged disk be used, it would restore the user to some single arbitrary point in the past. On the other hand, an external drive that receives changes chronologically from the main drive is capable of restoring to any number of points in time. **Thus after a crash, the external drive likely contains a safe point** followed by the transitional changes just preceding the crash. Since the transitional changes are useless, as they are incomplete, one reverts to the safe point.” *Id.*, 109:21-27

“Thus a guaranteed usable backup image is available, and depending on the lag in transferring changes, this point is likely not too far back in time. With a RAID system, protection is achieved from a **physical disk drive failure**, but none is provided for the **computer crashing** and leaving the last state of the disk in transition.” *Id.*, 109:28-110:2

“Providing for synchronization and continuous downloading of current image and historic information to an external hard disk in order to achieve a level of **hard disk redundancy**. The user can run from the external disk **should the main internal hard disk fail**. The **external disk is also used to re-initialize a new internal disk, after the failed disk is replaced**. This process is done concurrently with allowing the user to continue working.” EX1005, 130:24-29 (emphasis added).

Except for hard drive crash recovery, Schneider retrieves and/or recovers backup data from the history buffer from the computer’s local internal disk drive, not from a history buffer stored in an external disk drive. Schneider describes the Information Recovery Process in two different approaches. In the first case (in which a virtual drive D is created “by **combining information on the original drive C and the history buffer (which is typically part of drive C)**.” EX1005, 15:25-26. The process of looking back in time can involve the user setting a reference or target time for drive D, then, once drive D is created, accessing it as if it really was another physical disk drive whose contents had been copied from drive C at the specified time. Alternatively, the user’s main drive (i.e., “drive C”) is reverted to an earlier state by “copying the appropriate saved original data back into place, updating the history buffer to reflect the reversion, and restarting the system.” EX1005, 15:20-16:18; emphasis added.”

Schneider only uses the external disk to execute disk recovery in the event of a crash or hard drive failure. Schneider does not use the external disk to perform

selective recovery or for file recovery. Schneider's external disk drive is used only for crash recovery.

c. Schneider's file recovery embodiments

Although the majority of the Schneider reference is focused on describing its hard disk recovery technology, Schneider does contemplate approaches for using that technology for individual file recovery processes – i.e., recovering old versions of individual files.

On page 24 of the reference, for example, Schneider describes an approach for storing backup copies of files on a “circular system.” EX1005, 24:12-14. Although, as admitted by Schneider, this approach “may be more difficult to implement, prove correct, and isolate as a subsystem” (EX1005, 24:20-21) the reference contemplates that once in the so-called circular system, a user could “view what is in the circular history system and open/retrieve desired files.” EX1005, 25:6-7. This section of Schneider does not describe how to implement this file recovery process, let alone the process a user follows to retrieve a file from such a “circular system.”

Elsewhere in the reference, Schneider does describe some specific approaches for enabling a user to select a particular file to restore to trigger individual file recovery. For example, Schneider describes a method for file recovery in which “a new type of special ‘disk’ [is created] that can be examined through [Windows]

Explorer, where this disk does not correspond to any standard physical hard disk, but instead whose contents are generated based on file activity entries in the general log.” *Id.*, 60:8-11. The “file hierarchy for this special disk is formed by combining all relevant file entries currently found in the general log and sorting them.” *Id.*, 60:11-13. Browsing this special disk, the user can select a particular file to recover, in which case a “simulated disk” is generated, where the simulated disk includes a copy of the user’s hard drive from the time associated with the selected file. The file can then be recovered from that simulated disk. *Id.*, 59:11-17.

Accordingly, this embodiment of Schneider’s file recovery process relies upon the creation of a “simulated disk” from which individual files may be retrieved. As described above, in Section II.E.1.a, this process of creating a simulated disk (or simulated drive) in order to recover individual files relies on backup data stored locally to the user’s computer.

2. Crawford [EX1009]

Crawford (EX1009) is a patent directed to providing Internet-based computer services, such as virtual storage and compute services, to a user’s computer. EX1009, Abstract. Crawford contemplates that the remote virtual storage services could be useful for backup or file archiving purposes. In one instance, a user can configure the Crawford system for backup purposes by defining certain backup parameters, “Backup Option Flag, Incremental Flag, Full Backup, Backup File List

The backup option flag is used to indicate a backup request. Backup can be incremental, full backup or based on a file list. File list includes target location information.” EX1009, 33:2-6.

Crawford’s description of its backup capabilities is limited as the reference only mentions that that capability for full or incremental backups or file-specific backups exists. The references doesn’t discuss or appear to contemplate handling different file versions, the capability to browse and recover particular versions of backed-up files, or the idea of previewing a backed-up file before initiating full restoration of the file.

3. Sweat [EX1006]

Sweat (EX1006) is a patent assigned to Autodesk, Inc. Sweat describes a method and apparatus for drawing collaboration on a network. The Sweat patent application was filed on March 24, 2000, after the filing date of the ‘850 Patent, but claims priority to a provisional patent application filed on March 25, 1999. As such, Sweat is only a prior art reference to the ‘850 patent based on its priority claim to an earlier-filed provisional.

Sweat describes a “project hosting service” configured to enable multiple users to access and modify digital computer aided design (CAD) files, such as those used in architectural drawings. EX1006, Abstract.

As described by the reference, Sweat provides a “[m]ethod and apparatus for drawing **collaboration** on a network” and more specifically “a project hosting service that a user can access with a web browser” with which the user “can communicate and **collaborate** with the members of a design team or construction project” and “**share** CAD drawing files (DWG DWF, etc.), document files, or any other type of file.” EX1006, Title, Abstract (emphasis added).

Because Sweat describes a collaborative tool in which multiple engineers or architects may be working on the same set of drawings at the same time, Sweat’s system provides various mechanisms to control access to the digital files. For example, the system provides site administration functionality that enables “project and site administrators [to] control access and content for a site and project.” EX1006, 11:39-40. That control allows site administrators to “add members to projects or groups (who can also function as administrators (usually at the project level), if they are given the proper permissions); change name, password and email address information; change a member's company, title, and description information; and remove members from projects or groups.” EX1006, 11:44-49.

To enable control of how the various digital files are versioned, Sweat provides a specific “update” function that enables a user to version a file. At the time a file is versioned, the user can also provide information regarding the versioned file. As described by Sweat:

“An additional function that may be available from the version tab 508 is that of updating a file. If the user elects to update a file (e.g., by selecting the “Update” option from the right-click menu), the user is prompted to select the file on the local computer 104 that corresponds to the file 512 to be updated on server 106. The user can also elect to associate additional comments with the new version. Further, the user may have the option of notifying one or more other project members of the updated file (via email).” EX1006, 11:4-12.

As is well known in the engineering and architecture disciplines, when working collaboratively on CAD files, file versioning is an important process. In a large team, multiple members may be simultaneously working on different copies of particular versions of a particular set of drawings. As such, arbitrary changes to a file’s version number is to be avoided. For this reason, Sweat does not contemplate or attempt to provide a file backup solution that automatically backs-up files or file changes by uploading files or portions of files to Sweat’s central server as a new version of the file. Instead, file uploads to Sweat’s server must be triggered **manually** via the reference’s explicit “Update” function. Typically, updates are made with the submission of corresponding comments and description and, in some circumstances, automated communications being triggered to other members of the team. EX1006, 11:4-12.

Sweat does describe a mechanism by which a user can access the server to download particular versions of stored files but does not describe that download

process as being a “restoration” of the file. As described by Sweat, the system provides a file versions tab that “provides a list of all of the currently selected files. The versions tab 508 is used to open and view any file version, download a selected file to a local computer 104, and review or edit comments that are attached to the selected file version.” EX1006, 10:51-55. After reviewing the list of different file versions, the user can open a particular version of a file by selecting “the version from a list displayed in versions tab 508 and [selecting] a ‘View in Window’ button. Alternatively, the user can ... download a version of the file from the server 106 to the local computer 104.” EX1006, 10:56-59.

If the user wishes to create a new version of the downloaded file (e.g., to upload changes so that they can be accessed by other collaborators), the user must manually execute the “upload” function, as described above.

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

³ Emphasis added unless otherwise noted.

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

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

V. Argument

As detailed below, the Board should decline to institute review because the Petition fails to show a likelihood of success on the invalidity grounds presented in the Petition.

A. Ground 1: Schneider

a. Petitioner's Claim Mapping is Materially Deficient

Petitioner's claim mapping fails to "specify where each element of the claim is found in the prior art." 37 C.F.R. § 42.104(b)(4). 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 the majority of claim elements 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.

b. Claims 1, 10, and 18

- i. The Petitioner has failed to meet its burden in establishing "presenting information for a collection of one or more previous versions of the file ... wherein a restorable representation of each version, V, of the previous versions, is retrievable from a remote storage location...;" claim elements [1.2], [10.2], [18.2]**

Schneider does not describe presenting information relating to a collection of one or more previous versions of a file, wherein "a restorable representation of each version, V, of the previous versions, is retrievable from a remote storage location."

The Petitioner states that the “file-level data recovery implementation taught by Schneider maintains a restorable representation of each version presented to the user.” Petition, page 21. Specifically, the Petitioner states the “presented earlier file versions are restorable because such file versions are stored in the history buffer and are used to recover versions of desired files.” Petition, page 21.

However, Petitioner fails to establish where Schneider discloses storing such restorable representations **on a remote location**. In its so-called “file-level data recovery” implementations, Schneider only contemplates recovering files from a locally-stored repository – whether that’s the “circular system” of page 24 or the simulated disk of page 59. EX2001, ¶72.

As described above in Section II.E.1.b, although Schneider may contemplate storing historical buffer data on an “external disk,” that historical buffer data **does not** include “a restorable representation of each version, V, of the previous versions” of the file, as required by the claims. Instead, the data stored on the external disk consists of saved “disk states.” EX1005, 109:7-8. These disk states are not multiple “restorable representations” of different versions of a file. Instead, the disk states are records of logging activity that occurred on the user’s computer’s main disk. A POSITA would understand that saved disk states are not equivalent to “restorable representations” of versions of a file.

As described by Schneider, “instead of creating a backup of a specific point in time, the information on the backup includes the historic information, allowing the backup to re-create a range of ‘backup’ times. In other words, the external disk **generally mirrors the main internal disk.**” EX1005, 109:8-11. Using the external disk, therefore, “a guaranteed usable backup image is available.” EX1005, 109:28.

It makes sense that Schneider is focused on storing a disk image to its external disk rather than restorable representations of different versions of back-up files as Schneider teaches such an approach is preferred and can avoid a catastrophic loss of data. EX1005, 10:28-11:5. Moreover, in all of its file-recovery embodiments, Schneider retrieves individual files from **local** storage systems. EX2001, ¶76. When performing file recovery, it is presumed that the user’s computer is functioning and that the local hard disk is operational, and the history buffer stored thereon is accessible. As such, individual files can be retrieved from that local storage option. There is no need to access an external disk.

However, in the event of a hard disk failure, the external disk becomes necessary as no data can be recovered from the user’s local hard disk – the disk has failed. In that case, **the entire hard disk needs to be recovered**, in which case the disk images can be retrieved from the external disk and used to create a new instance of the user’s hard disk. Because writing data to the external disk may be laggy, Schneider acknowledges that the recovered hard disk data could be a little stale,

though “likely not too far back in time”, thereby discouraging such an implementation. EX1005, 109:29.

Schneider does not render remote individual file recovery obvious

In the Petition, the Petitioners argue that Schneider renders individual file recovery from a remote drive obvious because “a POSITA would have understood that such teachings were equally applicable to Schneider’s file level implementation because Schneider discloses that its ‘present invention can be implemented to work either with managing data at the disk sector level or the file level.’ Petition, page 22 (footnote 8).

In Schneider, however, the use of an external disk is presented as a solution to a specific problem – loss of the user’s hard drive. The reference provides no motivation to extend the use of that external disk to the specific file recovery case – Schneider already provides a better solutions (i.e., recovery from local storage) for file recovery. In fact, a number of factors including specific teachings of Schneider teach away and discourage a POSITA from the use of Schneider’s external disk for individual file recovery operations.

First, the external disk may contain a disk image that is out-of-date or stale. Due to slow transfer times and the potential of being disconnected for extended periods of time, Schneider acknowledges that the data contained in the external disk backup may be behind the internal disk drive backup or circular buffer. EX2001,

¶¶83-84. In other words, the external disk drive may not have the latest backup as the internal disk drive. This occurs because, in Schneider, the process of backing-up disk image data to the external disk differs from the process implemented by Schneider for its internal (local) backup operations. As described by Schneider, “The **external backup process of the present invention** differs from one in which the internal disk drive is simply copied onto another medium (e.g., disk or tape drive). Such a duplication is very time-consuming. Instead, the states of the external and internal drives are compared, and the appropriate historic and current image data is transferred, until both are synchronized. **This transfer process is asynchronous to and can lag substantially behind recent changes to the current image.** Therefore, it can be implemented on an inexpensive and relatively slow bus. For example, a parallel printer or USB port.” EX1005, 110:3-10 (emphasis added).

Further teaching away from the use of the Schneider’s external disk for file-retrieval operations is that accessing the external disk drive is slower than the internal disk drive. Why would a POSITA be motivated to use a slower mechanism for file recovery, when Schneider’s primary embodiment (storing backup data in a **local** circular buffer) is quicker and less likely to result in retrieval of stale data? EX2001, ¶¶85-86. As described by Schneider, the data transfer process to (and from) the external disk “is asynchronous to and can lag substantially behind recent changes to the current image. Therefore, it can be implemented on an **inexpensive and**

relatively slow bus. For example, a parallel printer or USB port.” EX1005, 110:7-10 (emphasis added).

Finally, further dissuading a POSITA from using Schneider’s external disk for file-recovery operations when file-recovery data is already available in **local** storage, is the fact that the external drive is optional and, at times, may be disconnected or unavailable. EX2001, ¶87. FIG. 47A of Schneider provides an illustration of how the external disk may be disconnected at any time: “Figure 47A illustrates **disconnected internal and external drives**. Each drive contains a current image and historic data. Initially the internal drive's four pages contain the values "A", "B", "C", and "D". **The external drive is blank**. In Figure 47B the values "a" and "b" are overwritten on "A" and "B", respectively. Thus, the original states move to the history buffer and the current image reflects the change. **The external drive is then connected** in Figure 47C. The engine responds by establishing a simulated disk based on the internal drive's current state (each write is assumed to also be a safe point). A dashed line in Figure 47C represents this time.” EX1005, 112:1-8 (emphasis added).

Because the external disk of Schneider is only used as a repository for data used to recovery a user’s entire hard drive, and because Schneider fails to teach or suggest the modification of the operation of that external disk to provide for individual file-recovery or versioning, Schneider fails to teach or suggest “wherein

a restorable representation of each version, V, of the previous versions, is retrievable from a remote storage location.”

**ii. The Petitioner has failed to meet its burden in establishing
“responsive to a selection to preview ... presenting a presentable
representation of the selected previous version” claim elements
[1.3], [10.3], [18.3]**

Schneider does not describe presenting a “presentable representation” of the “selected previous version” of a file or presenting that representation “responsive to a selection to **preview** the selected previous version.” As described above in Section II.C, this step enables a user to preview a particular version of a file to determine that it is the desired version, before restoring that version of the file as the current version.

In the Petition, the Petitioner argues that the present claim element is described by Schneider’s ability to open a restored file from a temporary directory. Specifically, the Petition argues that “[i]n response to user selection of a particular version, the Schneider system previews the version, e.g., by copying the previous file to a temporary directory and allowing the user to view that temporary copy.” Petition, page 28.

The Petition also states that the “previewing step” is found in Schneider’s ability to create, when performing file recovery, a “simulated disk” that contains the contents of the user’s hard drive and to open files from that simulated disk. The Petition refers to the section of Schneider that refers to opening files from a

simulated disk as provide a previewing function. Schneider describes that capability as follows:

“Note that it is useful to allow the user to select a file that can be retrieved from the past, and to automatically launch the appropriate application to view the file, referencing either the file on a simulated disk or copying the file from a simulated disk to a temporary directory. (The contents of this temporary directory, when no longer in use, are eventually automatically purged.) This allows the user to not only know of the existence of an old version of a file, but to view its contents without actually formally retrieving the file, as the viewed file is automatically discarded. Therefore, the viewed file's existence in terms of being retrieved is hidden from the user in that the user does not have to manage the viewed file on disk.” EX1005, 60:21-29.

But this capability of Schneider – to open recovered files from temporary folders – is not equivalent to a previewing function, as commonly understood. In Schneider, opening a file, whether in a temporary folder (i.e., a simulated disk) or elsewhere, requires that the file **has already been formally recovered**. There is no “presentable representation” of a file in Schneider, as contrasted with the “restorable representable.” In Schneider, there is no preview functionality. A file can only be opened/viewed if it has already been recovered, whether to a temporary folder or elsewhere.

If, as Petitioner states, the presentable representation is merely the same as the restorable representation, there is **no need** to retrieve the restorable representation

from the remote storage location because, in Schneider, once a file has been previewed (according to the Petitioner's reading), the file **has already been recovered** based upon available historical data. The file is already on the user's computer stored in the local "simulated disk."

Schneider, therefore, fails to teach or suggest "responsive to a selection to preview ... presenting a presentable representation of the selected previous version."

iii. The Petitioner has failed to meet its burden in establishing "responsive to a selection to restore the selected previous version, retrieving the restorable representation of the selected previous version from the remote storage location and storing the selected previous version ... as the current version on the local storage location" claim elements [1.4], [10.4], [18.4]

Schneider does not describe restoring the "selected previous version" by 1) "retrieving the restorable representation of the selected previous version **from the remote storage location**" and 2) "storing the selected previous version by the computational machine **as the current version** on the local storage location."

In the Petition, Petitioners refers to one of Schneider's "Practical Use" examples for its technology that allows for, "[h]ooking into or supplementing the directory and file presentation means of an OS, and allowing the user to view a list of earlier versions of a file. A selection can be made from the list and the recovered file either replaces the current version or is copied to a new file." EX1005, 130:10-13. The Petition then argues that based on this Practical Use statement a "POSITA would have understood this to involve retrieving the restorable representations from

the remote storage location and storing it locally because Schneider discloses restoring past files by replacing the current file with the recovered file.” Petition, pages 29-30.

As described above in Section V.A.b.i, however, this Practical Use of Schneider relies on its “file-recovery” mode of operation, which involves recovering file contents from a **local** storage location, such as Schneider’s circular buffer. Specifically, in its so-called “file-level data recovery” implementations, Schneider contemplates recovering files from a locally-stored repository – whether that’s the “circular system” of page 24 or the simulated disk of page 59. EX2001, ¶¶65-87. As discussed above in Section V.A.b.i, Schneider does contemplate, in certain circumstances, the use of an external disk to store disk image data, but that data and the external disk itself is only used to perform full hard drive recovery in the event of a disk failure. The external disk is not used for individual file recovery.

Consequently, Schneider does not teach or suggest “retrieving the restorable representation of the selected previous version from the remote storage location” as required by the claims.

Schneider similarly fails to teach or suggest “storing the selected previous version by the computational machine **as the current version** on the local storage location.” As recited by Schneider, once a file is recovered “the recovered file either **replaces** the current version or is copied to a new file.” EX1005, 130:12-13.

Schneider does not disclose storing the recovered file “as the current version” as required by the present claim. In Schneider, it appears that the act of recovering a file merely involves copying the (previously recovered) file from a temporary location to a more permanent location in which an existing file may be overwritten.

Consequently, Schneider fails to teach or suggest “responsive to a selection to restore the selected previous version, retrieving the restorable representation of the selected previous version from the remote storage location and storing the selected previous version by the computational machine as the current version on the local storage location.”

c. Claim 19

i. The Petitioner has failed to meet its burden in establishing “(d) previewing selected revision of said file, wherein said selected revision of said file is received from an Internet storage area network;”

Schneider does not describe 1) previewing a selected revision of a file or 2) where the selected revision of the file is received from an Internet storage area network.

As described above in Section V.A.b.ii Schneider does not describe a file preview capability.

As described above in Section V.A.b.iii Schneider does not describe receiving a revision of a file from an Internet storage area network. As discussed, Schneider relies on local repositories when performing file recovery.

- ii. The Petitioner has failed to meet its burden in establishing (f) previewing selected another revision of said file, wherein said selected another revision of said file is received from a network attached storage location;”**

Schneider does not describe 1) previewing a selected another revision of a file or 2) where the selected another revision of the file is received from a network attached storage location.

As described above in Section V.A.b.ii Schneider does not describe a file preview capability.

As described above in Section V.A.b.iii Schneider does not describe receiving a revision of a file from a network attached storage location. As discussed, Schneider relies on local repositories when performing file recovery.

B. Ground 2: Schneider+Crawford

Petitioner’s Ground 2 challenge also fails because it has not shown that a POSITA would have combined Schneider-Crawford in a manner that would render obvious the Challenged Claims.

Patent Owner submits that a POSITA would **not** have combined Schneider-Crawford in the claimed manner, at least because Petitioner’s own references teach away from the proposed combination and Petitioner’s purported motivations to combine the references are deficient.

Specifically, in Ground 2, Petitioner relies on Crawford for the element of remote storage location accessible via a network (independent claims 1, 10, 18), the

element of an internet storage area network (independent claim 19), and the element of network attached storage location (independent claim 19). For all other elements, Petitioners' Expert's Declaration relies on Schneider.

A POSITA would not have a combined Schneider-Crawford, 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 remotely stored back-up data, whether stored on an external disk or remote network drive for the purposes of file-level recoveries, and therefore Petitioner's proposed combination (relied on for all Claims in Ground 2) fails. EX2001, ¶¶128-136.

As described above in Section **II.E.1.b**, Schneider contemplates the use of an external disk to store disk images enabling recovery of an entire hard drive's contents in the event of disk failure. However, as described by Schneider, the use of an external disk is not optimal for file recovery because the contents of the external disk can differ (lag behind) the contents of the use's local hard drive:

"The **external backup process of the present invention** differs from one in which the internal disk drive is simply copied onto another medium (e.g., disk or tape drive). Such a duplication is very time-

consuming. Instead, the states of the external and internal drives are compared, and the appropriate historic and current image data is transferred, until both are synchronized. **This transfer process is asynchronous to and can lag substantially behind recent changes to the current image.** Therefore, it can be implemented on an inexpensive and relatively slow bus. For example, a parallel printer or USB port.” EX1005, 110:3-10; emphasis added.

In its file-recovery implementations, to mitigate these issues Schneider relies instead on data stored local to the user’s computer to perform file recovery - the data on the external disk is utilized to perform full disk recovery in the event of a disk failure.

“External Backup

All the methods thus far presented for saving original disk states are conceptually designed around a single disk. Of course, more than one disk may be involved, with their collective storage pooled into one large logical disk. The fault tolerance provided by the various methods deals with non-hardware failures like the user accidentally overwriting a file or a bug in an application corrupting files. However, there is also the **case of the disk actually ceasing to function (i.e., if it broke and the information it contained is lost)**. Recovery from such a failure typically involves installing a new hard disk, re-installing the operating system, and then restoring files from a backup tape or similar device.” EX1005, 108:15-24; emphasis added.

Consequently, the Schneider reference describes the use of external or remote backup capabilities for failures that require recovery of an entire hard drive. This may be necessary because in the event of a drive failure, local backups would be lost too – the external disk may be the only source of backup data. In the case of individual file recovery, however, the user's local hard drive is operating normally. As such, the backup data stored on that local drive is accessible and can be used for file retrieval, as taught by Schneider.

As evidenced by its external disk embodiment for drive recovery, Schneider was aware of the capabilities of remote and network storage devices. However, given the local nature of Schneider's data integrity solution, Schneider's describes only a local approach for file recovery which is most efficient given the ease and speed of accessing locally stored records. For file-recovery, Schneider has no reason to rely on accessing remote drives. Its preferred solution teaches away from such an approach. Why would a POSITA be motivated to modify Schneider to perform file-recovery operations from a remote or external disk, when Schneider's solution relies on recovery data that is stored locally when performing file recovery?

Consequently, a POSITA would have been discouraged from relying on any teachings of Schneider for the purposes of performing individual file-recovery via recovery of data from a remote data source. Schneider describes a solution

specifically configured to ensure that local copies of such records are available for the purposes of file recovery.

C. Ground 3: Sweat

1. Sweat is not Analogous Art

Sweat is directed to architectural drawing and project file sharing and collaboration, whereas the '850 Patent is directed to file preservation, file capture, file management, and file restoration. EX1001, Abstract, 1:12-16 (Field of the Invention).

Specifically, Sweat describes a collaboration system for multiple users to share project files and to distribute changes to files to other users working on the same project. EX1006, Title, Abstract. Using Sweat's system, a "user can communicate and collaborate with the members of a design team or construction project. [A] user can add comments or reply to discussion threads about the file or other aspects of the project. A user can view or download files from anywhere in the world to work on them, while locking the file to prevent others from overwriting the file." EX1006, Abstract.

Because Sweat describes a collaboration environment in which several team members may be modifying files and "checking-in" different copies of the files, the concept of a "current version" of any particular file is undetermined at any point in time. EX2001, ¶¶184-194. In the event that multiple users are working on the same

“version” of a file, it is impossible to determine which of those various files is the current version. As such, a POSITA would have understood from the teachings of Sweat that in order for a user to ensure that the user has the current version of a file on the user’s local disk drive, the user must perform a two-step sequence: (1) lock the file using the Sweat Lock command then execute Sweat’s upload functionality to (2) upload the latest version of the file from the user’s local disk drive to the Sweat server using the Sweat Update command. At that point, the file is uploaded and the version on the user’s disk drive as well as that on the server are the “current version” of the file. EX2001, ¶¶162-168.

In contrast, the ’850 Patent is not directed to allowing collaboration or file sharing between users.

“This information lets AccuSafe **keep the files from one user segregated from the files from another** use, since especially with large AccuSafe media, multiple users may be share a single AccuSafe storage device.” EX1003, p. 19; emphasis added.

Because the ’850 Patent describes a backup solution in which a user’s files are constantly being backed-up to remote storage as the user is making changes to the files, the use of the ’850 Patent technology in conjunction with a team of users making simultaneous edits to the same “file” would result in chaos. No user could have confidence that they are working on a “current version” of any file. As such, the technology of the ’850 Patent contemplates users working in isolation with their

own edits being uploaded for storage. And where recovery of a particular file version involves recovering an old version of a file such that the file, once downloaded, is designated the “current version” of that file, enabling those ongoing backup operations to continue. Specifically, the ’850 Patent is built on a platform that automatically archives files upon any operation on or changes to the files, ensuring that the remote storage always has the current version of any file from the local storage. As described in the present ‘850 Patent:

“The method includes, in a computing device, **detecting an instruction from a resident program to perform an operation on an operating file**. Upon detection of the instruction, **capturing the operating file temporally proximate to the operation** being performed on the operating file.” EX1001, 3:31-36; emphasis added

“File capture is preferably executed by **creating an archive file from the operating file**. The archive file is preferably stored in a temporary storage location, internal or external to the computer, such as input buffer 20. However, the archive file may be stored directly in storage device 35. In accordance with a preferred aspect of the invention, storage device 35 may be a personal attached storage device, a network attached storage device, an Internet storage area network, a peer-to-peer storage device, or other storage device.” EX1001, 5:31-40; emphasis added.

Sweat, therefore, describes a collaboration technology that is not analogous to the file archiving solution of the ‘850 Patent. Moreover, the proposed modification of Sweat’s system to provide the automated file archiving and recovery solution of the ‘850 Patent would render Sweat’s collaboration-based system unsatisfactory for its intended purpose and calls for significantly changing the principle of operation of Sweat.

Thus, because the Sweat reference is non-analogous art and the proposed modification to Sweat renders the reference inoperable, Petitioner’s Ground 3 necessarily fails.

- i. **The Petitioner has failed to meet its burden in establishing “responsive to a selection to restore the selected previous version, retrieving the restorable representation of the selected previous version from the remote storage location and storing the selected previous version ... as the current version on the local storage location” claim elements [1.4], [10.4], [18.3]**

Petitioner states that Sweat’s collaboration tool provides, for a given file, a “versions tab 508 [that] ‘is used to open and view any file version [and] download a selected file to a local computer.” Petition, page 55. The Petition states that the various file versions listed on the versions tab 508 would “also include the most recent, i.e., the current version of the file.” Petition, page 55. The user can then select one of the file versions listed in Sweat’s versions tab to download that version of the file.

At no time, however, does Sweat teach or suggest that the downloading of an *old* version of a file via the versions tab of Sweat causes that file, once downloaded to be the *current* version of the file on the local storage location. If the user downloads an old version of a file, Sweat does not contemplate changing the version associated with the file or setting the version of the file when it is stored in the user's computer. In Sweat, downloading a file is not equivalent to **recovering** the file. EX2001, ¶¶161-167.

If the act of downloading a file in Sweat caused that file to be designated as the current version of the file, that outcome could create significant confusion as the user works with other members of the collaboration team. The redesignation of an old version of a file as the current version of the file as part of the file restoration (in Sweat, just a simple file download) would be an undesirable outcome in the context of a collaboration tool.

As such, Sweat provides an entirely separate mechanism by which a user can designate a particular file as the “current version” of a file. EX2001, ¶164. Specifically, Sweat contemplates a manual “Update Function”:

Using that separate manual “Update Function”, “[i]f the user elects to update a file (e.g., by selecting the "Update" option from the right-click menu), the user is prompted to select the file on the local computer 104 that corresponds to the file 512 to be updated on server 106. The user can also elect to associate additional comments with the new version.

Further, the user 10 may have the option of notifying one or more other project members of the updated file (via email).” EX1006, 11:5-12.

Accordingly, in Sweat, the creation of a new version of a file (or the changing of a version of a file) is a deliberate and manual process that is accompanied by appropriate documentation and team notification.

In Sweat, the act of downloading a drawing file does not in any way affect the version designation for the file. Sweat’s collaborative nature teaches against such an implementation.

Consequently, a POSITA would understand that Sweat does not disclose “storing the selected previous version by the computational machine as the current version on the local storage location.”

D. Ground 4: Sweat+Schneider

In Ground 4, Petitioner addresses dependent claims 9 and 17. Petitioner fails to address the majority of claim elements of claim 9 and 17. To the extent that Petitioner relies on the claim chart that Petitioner presented in Ground 4 for the underlying elements of claim 9 and 17, Petitioner fails to carry its burden for claims 9 and 17 for the same reasons Petitioner failed to carry its burden for claims 1 and 10. *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 grounds asserted against the independent claims, [the Board] need[ed] to address only the grounds asserted against independent claims.”).

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
Datanet 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 10,364 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