UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

ACTIFIO, INC.
PETITIONER

V.

DELPHIX CORP.
PATENT OWNER

CASE IPR2015-00019
U.S. PATENT 8,150,808

PETITIONER ACTIFIO, INC.'S
NOTICE OF CROSS-APPEAL

Notice is hereby given, pursuant to 37 C.F.R. § 90.2(a), that Petitioner Actifio, Inc. ("Actifio") appeals to the United States Court of Appeals for the Federal Circuit from the Final Written Decision entered by the Patent Trial and Appeal Board (the "Board") on April 13, 2016 (Paper 69) (the "Final Written Decision," a copy of which is attached hereto).

In accordance with 37 C.F.R. § 90.2(a)(3)(ii), Actifio further indicates that the issue on appeal is the Board's decision that preponderant evidence failed to show that claims 27 and 52 of U.S. Patent No. 8,150,808 were obvious under 35 U.S.C. § 103(a) in light of prior art, along with all reasons, findings (including claim constructions), opinions, and orders leading thereto or underlying that decision, including, without limitation, the Board's failure to consider the effect of U.S. Patent No. 8,150,808 on the invalidating obviousness combination of the Edwards, Patterson, Sanders, and Fair references. Petitioner also notes that Case IPR2015-00019 has been appealed by Patent Owner Delphix Corp., and has been assigned Case No. 16-2065.

Simultaneous with this submission, a copy of this Notice of Cross-Appeal is being filed with the Patent Trial and Appeal Board, as well as with the Clerk of the United States Court of Appeals for the Federal Circuit.

Dated: May 24, 2016

By: /Robert Steinberg/

Robert Steinberg (Reg. No. 33,144)
Bob.Steinberg@lw.com
Latham & Watkins LLP
355 South Grand Avenue
Los Angeles, CA 90071-1560
213.485.1234; 213.891.8763 (Fax)

Counsel for Petitioner

CERTIFICATE OF SERVICE

I hereby certify that, on this 24th day of May, 2016, in addition to being filed electronically through the Patent Trial and Appeal Board's Patent Review Processing System (PRPS), the foregoing Petitioner Actifio, Inc.'s Notice of Cross-Appeal was sent via Express Mail to the following address pursuant to 37 C.F.R. § 104.2:

Office of the General Counsel
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

I further certify that, on this 24th day of May, 2016, a copy of the foregoing Petitioner Actifio, Inc.'s Notice of Cross-Appeal, along with the required docketing fee, was submitted electronically with the United States Court of Appeals for the Federal Circuit. One copy of Petitioner Actifio, Inc.'s Notice of Cross-Appeal was sent via Federal Express to the following address:

Clerk, U.S. Court of Appeals for the Federal Circuit
717 Madison Place, N.W.
Washington, D.C. 20439

I also certify that the foregoing Petitioner Actifio, Inc.'s Notice of Cross-Appeal was served via electronic mail on this 24th day of May, 2016, on the following attorneys designated by Patent Owner:

J. David Hadden (Reg. No. 40,629)
dhadden@fenwick.com

Saina S. Shamilov (Reg. No. 48,266)
sshamilov@fenwick.com
servicelist_delphixcounsel@fenwick.com
FENWICK & WEST LLP
Silicon Valley Center
801 California Street
Mountain View, CA 94041
Telephone: 650.988.8500
Facsimile: 650.938.5200

By: /Robert Steinberg/

Robert Steinberg (Reg. No. 33,144)
Bob.Steinberg@lw.com
Latham & Watkins LLP
355 South Grand Avenue
Los Angeles, CA 90071-1560
213.485.1234; 213.891.8763 (Fax)

Counsel for Petitioner

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

ACTIFIO, INC.,
Petitioner,

v.

DELPHIX CORP.,
Patent Owner.

Case IPR2015-00016 & IPR2015-00019[1]
Patent 8,150,808 B2

Before HOWARD B. BLANKENSHIP, KARL D. EASTHOM, and MINN CHUNG, *Administrative Patent Judges.*

CHUNG, *Administrative Patent Judge.*

FINAL WRITTEN DECISION
35 U.S.C. § 318(a) and 37 C.F.R. § 42.73

[1] As explained below, we hereby consolidate the two trials for purposes of issuing this Final Written Decision.

I. INTRODUCTION

In these *inter partes* review trials, instituted pursuant to 35 U.S.C. § 314, Petitioner Actifio, Inc. (“Petitioner”) challenges the patentability of certain claims of U.S. Patent No. 8,150,808 B2 (Ex. 1101, “the ’808 patent”), owned by Delphix Corp. (“Patent Owner”), as follows: claims 3, 29, 31, 36, 53, 54, and 56 in Case IPR2015-00016 (“’016 IPR”); and claims 2, 24–27, 35, 51, and 52 in Case IPR2015-00019 (“’019 IPR”). Based on a substantial overlap of arguments and evidence presented in the two cases, to administer the proceedings more efficiently, we exercise our authority under 35 U.S.C. § 315(d) to consolidate the two proceedings for purposes of issuing one final written decision.

The Board has jurisdiction under 35 U.S.C. § 6(c). This Final Written Decision is entered pursuant to 35 U.S.C. § 318(a) and 37 C.F.R. § 42.73. With respect to the grounds instituted in this trial, we have considered the papers submitted by the parties and the evidence cited therein. For the reasons discussed below, we determine Petitioner has shown by a preponderance of the evidence that claims 2, 3, 24–26, 29, 31, 35, 36, 51, 53, 54, and 56 of the ’808 patent are unpatentable, but has not shown by a preponderance of the evidence that claims 27 and 52 are unpatentable.

A. Procedural History

Based on Petitions (Paper 1 (“Pet.”) in the ’016 IPR; ’019 Paper 1 (“’019 Pet.”) in the ’019 IPR) filed by Petitioner, we instituted *inter partes* reviews of claims 2, 3, 24–27, 29, 31, 35, 36, 51–54, and 56 based on the following grounds of unpatentability (Paper 11 ("Inst. Dec."); '019 Paper 11 ("'019 Inst. Dec.")):

Claim(s) Challenged	Statutory Basis	Ground
2, 3, 24–26, 29, 31, 51, 53, 54, and 56	§ 103(a)	Edwards,[2] Patterson,[3] and Sanders[4]
36	§ 103(a)	Edwards, Patterson, Sanders, and Singh[5]
2, 27, 35, 51, and 52	§ 103(a)	Edwards, Patterson, Sanders, and Fair[6]

After institution of trial, Patent Owner filed a Patent Owner Response (Paper 23, "PO Resp."), to which Petitioner filed a Reply (Paper 32, "Pet. Reply").[7] Subsequently, Patent Owner moved to exclude (Paper 46, "PO Mot. to Exclude") certain Exhibits; Petitioner opposed (Paper 53, "Pet.

[2] Edwards et al., *FlexVol: Flexible, Efficient File Volume Virtualization in WAFL*, PROCEEDINGS OF THE ANNUAL TECHNICAL USENIX CONFERENCE 129–142 (June 22–27, 2008) ("Edwards") (Ex. 1103).

[3] Patterson et al., *SnapMirror®: File System Based Asynchronous Mirroring for Disaster Recovery*, PROCEEDINGS OF THE CONFERENCE ON FILE AND STORAGE TECHNOLOGIES, USENIX ASSOCIATION (January 28–30, 2002) ("Patterson") (Ex. 1104).

[4] Jawahar Lal & Roger Sanders, *DB2: Cloning a Database using NetApp FlexClone™ Technology*, Network Appliance Inc., IBM Toronto Lab, TR-3460 (Apr. 30, 2006) ("Sanders") (Ex. 1105).

[5] U.S. Patent No. 8,775,663 B1 (July 8, 2014) ("Singh") (Ex. 1106).

[6] U.S. Patent No. 7,334,095 B1 (Feb. 19, 2008) ("Fair") (Ex. 1206).

[7] Unless otherwise indicated, we refer to public (including redacted) Papers and Exhibits filed in IPR2015-00016.

Exclude Opp."); and Patent Owner replied (Paper 56, "PO Exclude Reply"). Patent Owner also filed Motions for Observation on certain cross-examination testimony of Dr. Erez Zadok (Paper 49, "Obs. Zadok") and Louis Hernandez (Paper 47, "Obs. Hernandez"), to which Petitioner filed Responses (Paper 57 ("Obs. Resp. Zadok") and Paper 55 ("Obs. Resp. Hernandez"), respectively). Patent Owner also filed a Paper identifying allegedly untimely evidence and evidence and arguments beyond the scope of Petitioner's Reply. Paper 58 ("Exclude Pet. Reply Evid."). The parties filed similar Papers and Exhibits in the '019 IPR (to be designated, as indicated above, with the "'019" prefix: for example, "'019 Paper").

A combined oral hearing in these proceedings and related Cases IPR2015-00014, IPR2015-00034, IPR2015-00025, IPR2015-00026, IPR2015-00050, IPR2015-00052, and IPR2015-00128 was held on January 14, 2016. A transcript of the hearing is included in the record as Paper 68 ("Tr.").

B. Related Proceedings

According to Petitioner, the '808 patent is the subject of the following pending patent infringement case: *Delphix Corp. v. Actifio, Inc.*, No. 5:13-cv-04613-BLF (N.D. Cal.). Pet. 2. In related proceedings before the Board, we instituted *inter partes* reviews of various claims of the '808 patent in Cases IPR2015-00014 and IPR2015-00034. Additionally, we instituted *inter partes* reviews of claims of U.S. Patent No. 8,161,077 B2 in Cases IPR2015-00025 and IPR2015-00026; claims of U.S. Patent No. 8,548,944 B2 in Cases IPR2015-00050 and IPR2015-00052; claims of U.S. Patent No.

8,566,361 B2 in Cases IPR2015-00100 and IPR2015-00108; and claims of U.S. Patent No. 8,468,174 B1 in Case IPR2015-00128.[8]

II. THE ’808 PATENT

A. Described Invention

The ’808 patent describes a system and method to create a virtual database, which involves obtaining multiple “point-in-time” (“PIT”) copies of the database to be virtualized. *See* Ex. 1101, Abstract. In one virtual database embodiment represented by Figure 2a, “production database system 110 . . . is the source of the database being virtualized” to create virtual database 220 using virtual database files stored in database storage system 100. *Id*. at col. 6, ll. 59–65. Figure 2a of the ’808 patent is reproduced below.

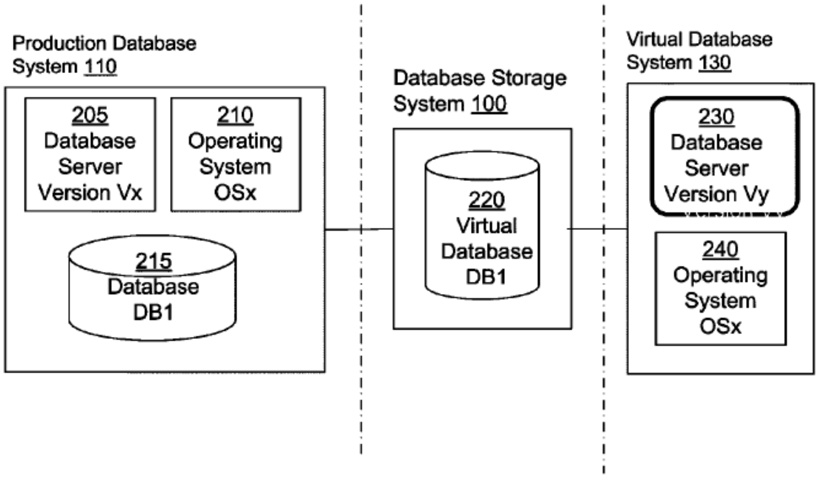

Production Database System 110
205 Database Server Version Vx
210 Operating System OSx
215 Database DB1
Database Storage System 100
220 Virtual Database DB1
Virtual Database System 130
230 Database Server Version Vy
240 Operating System OSx

FIG. 2(a)

Figure 2a depicts production database system 110, virtual database DB1 220 stored in database storage system 100, and virtual database system 130,

[8] Case IPR2015-00136 has been consolidated with IPR2015-00128.

which accesses virtual database 220.

To virtualize a production database, the system of the ’808 patent makes a first PIT copy of the production database and stores an entire set of database blocks representing the production database at that time in database storage system 100. *See* Ex. 1101, col. 18, ll. 27–36; Fig. 10. Subsequent PIT copies involve incremental changes and copy “only the blocks that changed since the last PIT copy and may copy much less data compared to the first PIT copy.” *Id*. at col. 18, ll. 38–41. A virtual database (VDB) is created by creating VDB file structures comprising VDB blocks that point to different PIT database blocks. *See id.* at col. 18, ll. 27–55. Each time an updated PIT copy is received at database storage system 100 reflecting changes in the production database, the system updates the appropriate VDB blocks in a VDB file which are “implemented as pointers to the actual database block that stores the [updated] data.” *See id*. at col. 18, ll. 44–55.

Figure 10 from the ’808 patent is reproduced below.

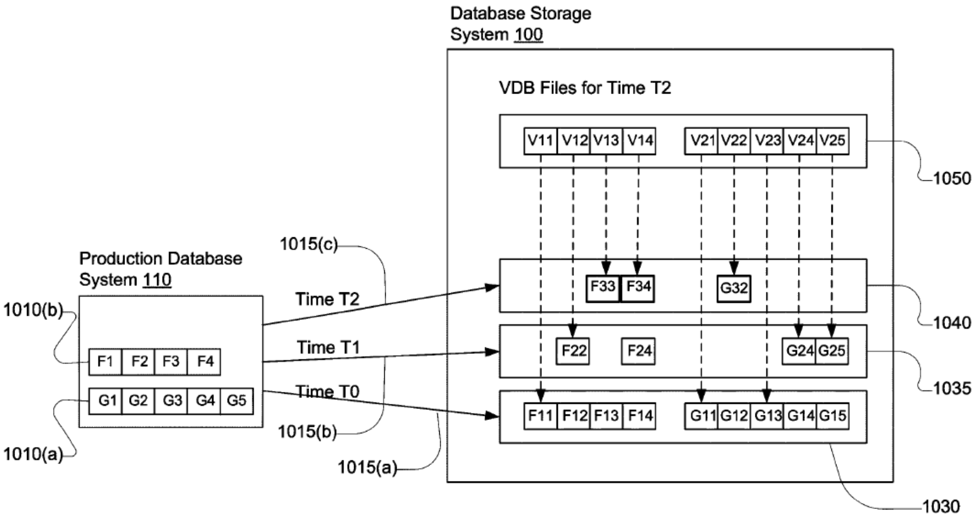


FIG. 10

Figure 10 shows "VDB Files for Time T2" in database storage system 100. Figure 10 further shows that "VDB file structures 1050" includes blocks V11, V12, V13, and V14 which point to database blocks F11 . . . F34 that represent different PIT (i.e., at times T0, T1, and T2) copies of production database blocks F1, F2, F3, and F4 at production database system 110. Initially, all the production database blocks are copied to create "[t]he first PIT copy 1030 made at time T0," as represented by database blocks F11, F12, F13, and F14 in database storage system 100. *Id*. at col. 18, ll. 35–38. Later, when the PIT copy made at time T2 is received and the VDB blocks are updated, block V13 points to the updated data at block F33, which represents a change existing at T2 to the data in block F3 in the production database (*see id*. at col. 18, ll. 53–55), whereas VDB block V11 still points to the data in block F11 "since the [production database] block F1 was never updated during copies made at time T1 and T2" (*id*. at col. 18, ll. 49–51).

B. Illustrative Claim

The challenged claims depend from claim 1 or 50.[9] Claim 1 is illustrative of the challenged claims and is reproduced below:

> 1. A method for creating a virtual database system, the method comprising:
>
> receiving different point-in-time copies of a source database, the source database comprising a plurality of database blocks;
>
> storing on a storage system, database blocks for a plurality of different point-in-time copies of the source database, wherein at least some of the stored database blocks

[9] Petitioner challenges claims 1 and 50 in IPR2015-00014.

> are associated with multiple point-in-time copies of the source database;
>
> creating a set of files for a virtual database, each file in the set of files is linked to the database blocks on the storage system associated with a point-in-time copy of the source database; and
>
> mounting the set of files associated with the virtual database on a database server allowing the database server to read from and write to the set of files.

III. CLAIM CONSTRUCTION

In an *inter partes* review, claim terms in an unexpired patent are given their broadest reasonable construction in light of the specification of the patent in which they appear. 37 C.F.R. § 42.100(b); *see also In re Cuozzo Speed Techs., LLC,* 793 F.3d 1268, 1277–79 (Fed. Cir. 2015) ("Congress implicitly approved the broadest reasonable interpretation standard in enacting the AIA," and "the standard was properly adopted by PTO regulation."), *cert. granted sub nom. Cuozzo Speed Techs., LLC v. Lee*, 136 S. Ct. 890 (mem.) (2016). In general, claim terms are given their ordinary and customary meaning in view of the specification, as would be understood by one of ordinary skill in the art at the time of the invention. *In re Translogic Tech. Inc.*, 504 F.3d 1249, 1257 (Fed. Cir. 2007); *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312–13, 1315 (Fed. Cir. 2005) (en banc). A patentee may rebut that presumption by providing a definition for the term in the specification with reasonable clarity, deliberateness, and precision. *In re Paulsen*, 30 F.3d 1475, 1480 (Fed. Cir. 1994). A particular embodiment appearing in the written description generally is not incorporated into a claim if the claim language is broader than the embodiment. *SuperGuide*

Corp. v. DirecTV Enterprises, Inc., 358 F.3d 870, 875 (Fed. Cir. 2004); *In re Van Geuns*, 988 F.2d 1181, 1184 (Fed. Cir. 1993); *see also Phillips*, 415 F.3d at 1323 ("[A]lthough the specification often describes very specific embodiments of the invention, we have repeatedly warned against confining the claims to those embodiments.").

During trial, the parties disputed the claim construction of the terms "database block" and "virtual database," which we address below. No other claim terms require express construction to resolve the issues raised in this *inter partes* review. *See, e.g., Vivid Techs., Inc. v. Am. Sci. & Eng'g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999) (only those terms that are in controversy need to be construed, and only to the extent necessary to resolve the controversy).

The parties' claim construction arguments advanced in Case IPR2015-00016 are essentially identical to their arguments presented in Case IPR2015-00019. Hence, although our discussion below focuses on the '016 IPR, our findings and conclusions apply equally to the '016 IPR and the '019 IPR cases unless indicated otherwise.

A. Database Block

1. Whether a Database Block Requires Metadata

The main claim construction dispute between the parties with respect to the term "database block" centers on whether a database block must necessarily include metadata. Patent Owner asserts the term "database block" should be interpreted to require metadata, i.e., as "a unit of data used by a database which comprises a specific number of bytes stored in the storage, a portion of which stores metadata associated with the unit of data" (PO Resp. 16), whereas Petitioner argues the correct interpretation of the term is not so limited, i.e., "a unit of data used by a database" (Pet. 10). For the reasons discussed below, we conclude the disputed term is not limited as Patent Owner contends.

a. Claim Language

We begin our claim construction analysis by considering the language of the claims themselves. *Phillips,* 415 F.3d at 1314. First, we note that the term "metadata" is not recited in any of the challenged claims. Nor do the claims expressly require inclusion of metadata in database blocks. The only claims of the '808 patent that recite "metadata" are dependent claims 32 and 33, which are not challenged in this case or any other related cases currently before the Board. These claims depend indirectly from claim 1 and expressly recite "metadata of database blocks." Thus, had the patentee intended to limit "database blocks" recited in claim 1 or any other challenged claims to require metadata, it could have done so by explicitly modifying the disputed term with "metadata," but did not.

Therefore, to show the disputed term is limiting, Patent Owner must demonstrate "a clear indication in the intrinsic record that the patentee intended the claims to be so limited." *Liebel–Flarsheim Co. v. Medrad, Inc.*, 358 F.3d 898, 913 (Fed. Cir. 2004); *see also Aventis Pharma S.A. v. Hospira, Inc.*, 675 F.3d 1324, 1330–32 (Fed. Cir. 2012) ("perfusion" not limited to having at least eight hours of stability because the patentee did not "clearly express an intent" to redefine the term in the specification or during prosecution).

b. Written Description

Turning to the written description, Patent Owner asserts that the following passage in the Summary section of the Specification defines the term "database block":

> A database block is a unit of data used by a database and comprises a specific number of bytes stored in the storage. A database block can also be referred to as a page. A portion of the database block stores metadata associated with the database block.

PO Resp. 16 (quoting Ex. 1101, col. 2, ll. 7–12). The first sentence in the cited passage above explicitly defines the term, by stating "[a] database block is a unit of data used by a database."[10] Ex. 1101, col. 2, ll. 7–9. Although the second next sentence states that a database block "stores metadata," that sentence by itself is insufficient to limit the disputed term by requiring the unrecited "metadata" feature because it does not state unambiguously that all "database blocks" must include metadata. *See Liebel-Flarsheim*, 358 F.3d at 906 (construing a claim term broadly because "[n]o statement in the written description [ ] constitute[d] a limitation on the scope of the invention") (quoting *Brookhill-Wilk 1, LLC v. Intuitive Surgical, Inc.*, 334 F.3d 1294, 1301 (Fed. Cir. 2003)). Further, the cited passage does not exclude the possibility of some database blocks not having any metadata. *See id.* at 908 (passages in the Summary of the Invention

[10] The first sentence also states a database block comprises "a specific number of bytes stored in the storage." For the reasons discussed in Section III.A.4 below, we find this addition is not part of the explicit definition but, rather, represents embodiments within the defined term.

section of a patent did not limit the scope of the invention because the passages, "although focusing on the use of the invention in conjunction with pressure jackets, do not disclaim the use of the invention in the absence of a pressure jacket"). Nonetheless, if the rest of the Specification, e.g., the Detailed Description section, clearly and consistently describes the claimed invention as requiring metadata in database blocks, such a limiting description together with the sentences cited above may support a limiting construction of the disputed term. *Compare Am. Piledriving Equip., Inc. v. Geoquip, Inc.*, 637 F.3d 1324, 1333–34 (Fed. Cir. 2011) (citing *C.R. Bard, Inc. v. U.S. Surgical Corp.*, 388 F.3d 858, 865–66 (Fed. Cir. 2004)) (holding that a limiting description in the specification supports a limiting construction of a claim term when the limiting feature is referenced "throughout the specification," and "other statements and illustrations in the patent are consistent with the limiting description"), *with MEMS Tech. Berhad v. Int'l Trade Comm'n*, 447 F. App'x 142, 151 (Fed. Cir. 2011) (nonprecedential) (distinguishing *C.R. Bard* and finding the statements in the abstract and summary sections to be non-limiting because, in *C.R. Bard*, the specification *universally* describes a limiting feature of the invention whereas in *MEMS*, "the general language in the abstract and summary sections does not represent the full scope of the embodiments in the specification"). In this case, as discussed below, our review of the Specification, including the portions identified by Patent Owner, does not reveal a limiting description sufficient to support a limiting construction.

Patent Owner asserts, citing certain portions of the Specification and the Declaration of Prashant Shenoy, Ph.D. (Ex. 2111, "Shenoy Decl."), that if a database block does not include metadata, the system disclosed in the ’808 patent would not work as described. PO Resp. 16–18. For example, Patent Owner argues that a database block must include metadata because the disclosed system analyzes the metadata of each block to store only the incremental changes made to the production database (*id.* (citing Ex. 1101, col. 13, ll. 34–46, 46–51)), which is “one of the main functions” of the claimed system (*id.* at 17 (citing Ex. 1101, col. 6, ll. 34–46, 43–46, col. 7, ll. 49–57; Ex. 2111 ¶¶ 76, 79–80)) essential to achieving “a main purpose” of the invention—“to efficiently provide virtual database . . . without proliferating redundant copies of database data” (*id.* (citing Ex. 2111 ¶¶ 63–67)). Patent Owner also asserts that metadata is required in each database block in order to map the block to a database file and a location within that file. *Id.* at 18 (citing Ex. 1101, col. 14, ll. 27–31; Ex. 2111 ¶¶ 76, 81).

Patent Owner’s argument is unpersuasive because the argued advantages or purposes are not recited features of the claims. Moreover, a claim is not required to encompass all of the advantages or purposes of the invention. *See Howmedica Osteonics Corp. v. Wright Med. Tech., Inc.*, 540 F.3d 1337, 1345 (Fed. Cir. 2008) (“An invention may possess a number of advantages or purposes, and there is no requirement that every claim directed to that invention be limited to encompass all of them.”) (citations omitted).

Furthermore, the disclosure in the passages cited by Patent Owner above is not limiting because the passages describe a particular embodiment of making a point-in-time copy of the production database by streaming data to the database storage system, where the data stream is formatted to include metadata in each transmitted database block. As noted by Petitioner (Pet. Reply 8), the ’808 patent discloses an alternative embodiment where the transfer of production database data is achieved by “using a file sharing system similar to the file sharing system 120” (Ex. 1101, col. 7, ll. 57–64), such as a network file system (NFS) (*id.* at col. 10, ll. 35–37). As discussed below, there is no disclosure in the written description that requires metadata in each database block used in the file sharing embodiment.

In “the streaming embodiment,” which is depicted in Figures 4 and 5 and described in column 12, line 14 to column 14, line 67 of the ’808 patent (*see id.* at col. 3, ll. 28–37 (describing Figures 4 and 5 as “an embodiment of the invention”)), the production database system, upon receiving a request for data from the point-in-time copy manager of the database storage system (*id.* at col. 12, ll. 19–23), packages the production database data “into a format that can be processed by the point-in-time copy manager” (*id.* at col. 12, ll. 58–62) and builds the appropriately formatted data into a data stream that is sent to the point-in-time copy manager. *Id.* at col. 12, l. 62–col. 13, l. 3. Upon receiving the data stream, the point-in-time copy manager processes the data stream to identify database blocks contained in it. *Id.* at col. 13, ll. 27–33. In the data stream, “[e]ach database block includes metadata” (*id*. at col. 13, ll. 33–34), which is used, for example, to “identify database block boundaries in the stream of data” (*id*. at col. 14, ll. 22–25). When saving a retrieved database block into a transferred or copied database file on the database storage system, the point-in-time copy manager “analyzes the database block metadata to map [] the database block to [the] database file and a location within the file.” *Id.* at col. 14, ll. 29–31.

We find nothing in this disclosure regarding the streaming embodiment that limits the claimed invention as Patent Owner contends. For example, it may be necessary to include metadata in each database block transmitted on a data stream in order to identify and unpack database blocks from a continuous stream of data that has no apparent structure or boundaries. But this does not show that the same approach is necessary in a file sharing embodiment where the database files to be copied have defined boundaries and known structures, and the database blocks stored in the files can be accessed directly. *See, e.g., id.* at col. 13, ll. 10–12 ("the production system library [] includes code to analyze *the structures of the files* of the database stored in the data store and *also includes code to process metadata associated with database blocks* stored in the data store") (emphasis added);[11] col. 6, ll. 11–17 ("FIG. 1 illustrates one embodiment for how information may be *copied* from a production database to a database storage system . . . *using a file sharing system*. . . . In some embodiments information may be *copied* from *storage level snapshots* of production databases.") (emphases added). Further, the fact that the streaming embodiment uses metadata to map the database blocks unpacked from a data stream to a copied database file for storage does not require database blocks used in the file sharing embodiment to have metadata because, when file sharing is used, the database file on the production system can be accessed and copied directly by "mounting the production DB data store" on the database storage system (*id.* at col. 7, ll. 57–64) without packing and

[11] The phrase "metadata *associated* with database blocks" implies any metadata need not be *in* the database blocks.

unpacking the database blocks of the database file into and out of data streams.

Furthermore, contrary to Patent Owner's argument, analyzing the metadata of each database block in the data stream is not necessary to achieve incremental updates because the passages cited by Patent Owner describe only one of the two embodiments disclosed in the '808 patent for achieving the incremental copy function. In the embodiment relied upon by Patent Owner, the data stream may include unnecessary database blocks, such as the blocks that did not change since the last point-in-time copy was transmitted, which are eliminated after the data stream is received at the database storage system by analyzing metadata for each database block. *Id.* at col. 13, ll. 43–64. In an alternative embodiment, which is not addressed by Patent Owner, the unchanged blocks are eliminated at the production system and never sent to the database storage system. *Id.* at col. 13, l. 64–col. 14, l. 3 ("In *other* [sic] *embodiment*, some or all of the unnecessary blocks may be *eliminated while the data stream is built* by the production system library []. In this embodiment, the data stream . . . is reduced in size resulting in efficient communication between [the production system and the database storage system].") (emphases added). Patent Owner does not explain why metadata must be included in each database block to achieve the incremental update function in this embodiment. Hence, packing metadata within database blocks may be involved in some streaming embodiments, but nothing in the Specification indicates it is required for the incremental update function. Therefore, there is nothing in the Specification that indicates that copying database files by streaming data is the essence of the claimed invention rather than a preferred embodiment, which may not be read into the claims "absent clear disclaimer in the specification." *In re Am. Acad. of Sci. Tech Ctr.*, 367 F.3d 1359, 1369 (Fed. Cir. 2004).

c. Other Intrinsic Evidence

Considering "the context of the surrounding words" to the term "database block" in the claims, which "must be considered in determining the ordinary and customary meaning" of the disputed term, *ACTV, Inc. v. Walt Disney Co.*, 346 F.3d 1082, 1088 (Fed. Cir. 2003), independent claims 1, 50, and 57 recite receiving point-in-time copies of a source database and storing database blocks associated with the received point-in-time copies on a storage system. But the claims do not say anything about a particular method of transferring point-in-time copies, whether by streaming or by file sharing. Hence, the claims cannot be limited to either embodiment, and, therefore, the streaming embodiment cited by Patent Owner does not limit the claims.

In addition, consideration of differences among the claims of the '808 patent supports the conclusion that the passages Patent Owner cited are not limiting. Claim 29, which depends from claim 1, recites "receiving point-in-time copies" by "receiving data streams" which comprise "data from database blocks." Claims 32, 33, and 34 each depend from claim 29 and additionally recite "identify[ing] database blocks" in the data streams, "analyzing the metadata of database blocks to determine the length of the database blocks" (claim 32), "analyzing the metadata of database blocks to determine whether the database block needs to be stored" (claim 33), and "determining not to store the database blocks that . . . did not change since a previous retrieval of point-in-time copy" (claim 34). Hence, the subject matter specifically claimed in these dependent claims correspond to the written description in the Specification relating to the streaming embodiment discussed above. *See* Shenoy Decl. (Ex. 2111) ¶ 80; PO Resp. 17 (citing Shenoy Decl. ¶ 80).

Claim 1, from which these claims depend, is presumed to be broader and not limited by the additional limitations relating to streaming recited in these dependent claims. "[I]n a situation where dependent claims have no meaningful difference other than an added limitation, the independent claim is not restricted by the added limitation in the dependent claim." *Trustees of Columbia Univ. v. Symantec Corp.*, 811 F.3d 1359, 1370 (Fed. Cir. 2016) (citing *Phillips,* 415 F.3d at 1314–15; *Acumed LLC v. Stryker Corp.*, 483 F.3d 800, 806 (Fed. Cir. 2007)). Therefore, claim 1 cannot be read, absent other evidence, to be limited to the streaming embodiment, and, accordingly, the passages cited by Patent Owner relating to the streaming embodiment do not limit claim 1 to require a "database block" to have "metadata." *See Columbia Univ.*, 811 F.3d at 1370 (holding that, in the absence of rebutting evidence, a disputed term recited in claim 1 cannot be read to be limited to use only the type of data recited in dependent claims because the dependent claims are presumed to be narrower than the independent claims on which they depend); *see also Toshiba Corp. v. Imation Corp.*, 681 F.3d 1358, 1368–69 (Fed. Cir. 2012) (construing claim 1 to read on both single-sided and double-sided discs when the plain language of the claim is broad and a dependent claim added a requirement specifically reciting the number of disc sides).

Patent Owner's argument is deficient in another aspect—namely, its failure to address database blocks outside the context of data streams. The database blocks transmitted in a data stream described in the passages cited by Patent Owner are in transit between the production system and the database storage system. But, as discussed above, the challenged independent claims recite, in addition to receiving point-in-time copies of a database, storing database blocks on a storage system. Patent Owner does not cite, nor do we discern, anything in the Detailed Description section of the Specification that requires metadata in database blocks that are stored in storage—that is, database blocks stored in the production database system before being packaged and formatted into the data stream or stored in the database storage system after being unpacked from the received data stream. Contrary to Patent Owner's contention, the written description in fact suggests metadata need not be included in stored database blocks. For example, the '808 patent describes storing database blocks unpacked from data stream as follows: "The file in which the database block is saved comprises a file header including metadata associated with the file and a sequence of database blocks." Ex. 1101, col. 14, ll. 44–47. This passage suggests that the metadata can be stored in the file header separately from the series of database blocks stored in the body of the file. In sum, we find no disclosure in the Detailed Description section of the Specification that clearly and consistently describes the claimed invention as requiring metadata in all database blocks.

Therefore, in view of the entire disclosure of the ’808 patent and the plain language of the claims, we find, notwithstanding the statement in the Summary section relied upon by Patent Owner—“[a] portion of the database block stores metadata associated with the database block”—the intrinsic record does not justify limiting the term “database block” by reading in the “metadata” limitation not found in the claims. *See Liebel-Flarsheim*, 358 F.3d at 908; *MEMS*, 447 F. App’x at 151.

d. Extrinsic Evidence

Patent Owner also argues additional evidence supports a limiting construction. For example, citing the testimony of Dr. Shenoy and the testimony of Petitioner’s expert, Dr. Zadok (Ex. 1119, “Zadok Decl.”), Patent Owner asserts that all database management systems mentioned in the ’808 patent, such as Oracle and IBM DB2 (Ex. 1101, col. 5, ll. 4–8), require metadata in database blocks. PO Resp. 20–21 (citing Ex. 2111 ¶¶ 41–46; Ex. 1119, 56 n.13). In the paragraphs cited by Patent Owner, Dr. Shenoy discusses various documents describing the database systems listed in the ’808 patent, including Oracle, Sybase, Microsoft SQL Server, and IBM DB2, and testifies that these database systems all require metadata in database blocks.[12] Ex. 2111 ¶¶ 43–46. The evidence presented by Patent Owner in support of its argument—i.e., testimony of experts and documents describing the commercially available database systems listed in the ’808 patent—is more properly characterized as extrinsic evidence. Such extrinsic

[12] However, neither Patent Owner’s brief nor Dr. Shenoy discusses the MYSQL database system mentioned in the ’808 patent.

evidence is "less significant than the intrinsic record in determining the legally operative meaning of claim language." *Phillips*, 415 F.3d at 1317 (citations omitted) (internal quotation marks omitted). Even if the external documents discussed by Dr. Shenoy are deemed to describe embodiments of the '808 patent, the evidence would be insufficient to limit the term "database block" because reading in the "metadata" limitation not found in the claims from preferred embodiments is improper. *See Cadence Pharm. Inc. v. Exela PharmSci Inc.*, 780 F.3d 1364, 1369 (Fed. Cir. 2015) ("[E]ven if *all of the embodiments* discussed in the patent included a specific limitation, it would not be proper to import from the patent's written description limitations that are not found in the claims themselves.") (emphasis added) (citations omitted) (internal quotation marks omitted); *see also Dealertrack, Inc. v. Huber*, 674 F.3d 1315, 1322 (Fed. Cir. 2012) ("The disclosure of multiple examples does not necessarily mean that such list is exhaustive or that non-enumerated examples should be excluded."). Moreover, Petitioner points to testimony from Dr. Zadok contradicting Dr. Shenoy's testimony. Pet. Reply 9–10 (citing Ex. 1166 ¶¶ 13–20). Dr. Zadok explains that at least Oracle and IBM DB2 databases also include database blocks *without* metadata and concludes that "[t]he ordinary meaning of the term 'database block' does not include metadata." Ex. 1166 ¶ 13.

Relying upon the testimony of Dr. Shenoy, Patent Owner also asserts that "it is well known to one skilled in the art that a database block necessarily includes metadata." PO Resp. 21 (citing Ex. 2111 ¶¶ 41–46); *see also id.* at 16 ("[T]hat database blocks include metadata is consistent with the ordinary meaning of the term in the database field and comports with how the term is used by every major database system provider.") (citing Ex. 2111 ¶¶ 41–46, 74). In his Declaration, in addition to the documents describing various commercially available database systems, Dr. Shenoy also discusses a treatise on database systems (Ex. 2108, "Molina") and testifies that it is generally understood that a database block or a page will include metadata. Ex. 2111 ¶ 42 (citing Ex. 2108, 29, 31). The portion of the Molina treatise cited by Dr. Shenoy, however, appears to describe features of a *relational* database system, not characteristics common to all databases in general. *See, e.g.*, Ex. 2108, 29 ("Collections such as *relations* are usually represented by placing the records that represent their data elements in one or more blocks.") (emphasis added), 31 ("Records representing tuples of *a relation* are stored in blocks of the disk . . . there is a *block header* holding information such as: . . . Information about which *relation* the tuples of this block belong to.") (first and last emphases added).

In fact, Petitioner argues that Patent Owner does not provide any evidence that database blocks require metadata in database systems other than relational database systems. *See* Pet. Reply 10. Petitioner correctly points out that the '808 patent Specification states that the disclosed invention "can be used for *any database*." *Id*. (quoting Ex. 1101, col. 5, ll. 13–15) (emphasis by Petitioner). Indeed, the '808 patent states that "[a] database may be implemented using a database model, for example, a relational mode, object model, hierarchical mode or network model" and goes on to state that "the techniques disclosed can be used for any database." Ex. 1101, col. 5, ll. 8–15. Thus, we agree with Petitioner that the '808 patent does not limit its disclosed invention to relational database technology.

Relying on the testimony of Dr. Zadok submitted with its Reply (Ex. 1166, "Supp. Zadok Decl."), Petitioner asserts that, in other types of database systems, such as Google's BigTable database, metadata is stored separately from the database blocks.[13] Pet. Reply 10 (citing Ex. 1151, 4; Ex. 1166 ¶ 22). Petitioner also argues, citing the testimony of Dr. Zadok and another treatise on database systems (Ex. 1152, "Elmasri"), that a "flat file database," such as a comma separated value (CSV) file used by spreadsheet applications, does not have metadata in its blocks. Pet. Reply 11 (citing Ex. 1166 ¶¶ 23–24, Ex. 1152, 15). Therefore, Petitioner asserts that the ordinary meaning of database block does not require including metadata. *See id.,* Ex. 1166 ¶ 25.

We agree with Petitioner that the extrinsic evidence Patent Owner relies upon to argue for a limited ordinary meaning rests on relational databases. Because the challenged claims plainly recite "database," and, therefore, are not limited to relational databases, Patent Owner's evidence,

[13] Petitioner argues the definition of a "database" proposed by Patent Owner's expert, "a collection of data that is organized so that it can be easily accessed, managed or updated," encompasses many types of databases other than the relational databases. *See* Pet. Reply 10 (citing Ex. 2111 (Shenoy Decl.) ¶ 33). In addition, the '808 patent describes broadly that "[a] database comprises data stored in a computer for use by computer implemented applications." Ex. 1101, col. 4, l. 67–col. 5, l. 1. Under either the '808 patent's description or Dr. Shenoy's testimony, we agree with Petitioner that a "database" encompasses many types of databases other than the relational databases, including Google's BigTable, spreadsheet databases, and flat files.

even assuming it shows most or all relational databases have database blocks that require metadata, does not establish an ordinary and customary meaning of "database blocks" recited in the claims. Therefore, we find Patent Owner's extrinsic evidence regarding ordinary meaning does not overcome the intrinsic record of this case. *See Phillips*, 415 F.3d at 1318 ("[A] court should discount any expert testimony that is clearly at odds with the claim construction mandated by the claims themselves, the written description, and the prosecution history, in other words, with the written record of the patent.") (citation omitted) (internal quotation marks omitted). Furthermore, we are persuaded by Petitioner's evidence that certain well-known databases, such as Google's BigTable database, do not require metadata in database blocks, and, therefore, find that the evidence in this case does not establish a widely accepted meaning of "database block" in the field of database systems that requires metadata to be necessarily included in a database block.

Moreover, notwithstanding the competing extrinsic evidence regarding ordinary meaning from the parties, our focus in claim construction must properly remain with the written description and the language of the claims. *See Kara Tech. Inc. v. Stamps.com Inc.*, 582 F.3d 1341, 1348 (Fed. Cir. 2009) ("It is not uncommon in patent cases to have [] dueling experts. When construing claims, however, the intrinsic evidence and particularly the claim language are the primary resources."). Upon weighing the competing extrinsic evidence regarding ordinary meaning from the parties and in view of our analysis of the written description and claim language discussed above, we find Patent Owner's extrinsic evidence is not sufficient to overcome the plain language of the claims and, therefore, decline to read the “metadata” limitation into the term “database block.” *See id.* (finding the testimony of an expert cannot overcome the plain language of the claims and rejecting a proposed construction that limits a claim term by reading in a limitation not recited in the claims); *see also Phillips*, 415 F.3d at 1321 (“Properly viewed, the ‘ordinary meaning’ of a claim term is its meaning to the ordinary artisan after reading the entire patent.”).

2. “Application Level” vs. “Storage Level” Distinction

Patent Owner next argues that a “database block” must include metadata because “the ’808 patent describes an *application level* system that understands database blocks and uses metadata in those database blocks to determine whether to store database blocks and where to store them.” PO Resp. 26 (emphasis added); *see also id.* at 14 (making the same argument). Patent Owner asserts, therefore, a “database block” cannot be a “*storage level* unit of data.” *Id.* at 23, 26–27 (emphasis added). Patent Owner cites passages in the ’808 patent in support of its argument. *Id.* at 14 (citing Ex. 1101, col. 13, ll. 27–41, 43–60). The cited passages, however, describe processing of database blocks in a stream of data in the streaming embodiment and, therefore, do not limit the challenged claims as discussed above. Furthermore, the cited passages relate to a particular method, i.e., data streaming, of transferring or copying the production database data to the storage database system—an alternative method being data transfer by file sharing—and do not relate to whether a “database block” is an “application level" or "storage level" entity or concept. Therefore, Patent Owner's argument is unpersuasive.

Patent Owner also asserts that the system of the '808 patent "operates at the application level" because the system uses APIs (application program interfaces) to copy database blocks. *Id.* at 14 (citing Ex. 1101, col. 7, ll. 41–52). Petitioner argues that Patent Owner's argument is contrary to disclosed embodiments of the '808 patent because the patent teaches the system "may retrieve the necessary database blocks from *storage level snapshots* of production databases." Pet. Reply 12 (citing Ex. 1101, col. 8, ll. 33–37) (emphasis by Petitioner). We agree with Petitioner's argument and further note that the '808 patent also discloses copying of database blocks from *storage level* snapshots of a database. *See* Ex. 1101, col. 6, ll. 11–17 ("FIG. 1 illustrates one embodiment for how information may be *copied* from a production database to a database storage system . . . using a file sharing system. . . . In some embodiments information may be *copied* from *storage level snapshots* of production databases.") (emphases added). More importantly, Patent Owner does not explain why the fact that the system of the '808 patent uses APIs to access database blocks necessarily requires the database blocks to include metadata. For example, Patent Owner does not identify, nor do we discern, anything in the written description that explains why the APIs cannot access and process metadata that is stored separately from database blocks.

Patent Owner further relies upon the testimony of Dr. Shenoy and external documents to argue that a "database block" is an "application level" entity, which is different from a "file system block," an "operating system block," or a "storage level" unit of data. PO Resp. 23–27 (citing Ex. 2111 ¶¶ 47, 122–35). In particular, Patent Owner asserts that an Oracle document (Ex. 2102, "Oracle Manual") explains this "well-known distinction." *Id.* at 23–24 (citing Ex. 2102, 250, Fig. 12-5).[14] Patent Owner further argues "neither are file system blocks accessible to a database nor are database blocks accessible to the file system." *Id.* at 26 (citing Ex. 2111 ¶¶ 122–35). Therefore, Patent Owner argues a construction of the term "database block" that encompasses a storage unit block is incorrect. *Id.* at 27.

As a threshold matter, we note that the Oracle Manual document is dated May, 2014, which is several years later than the filing date of the '808 patent. Hence, the Oracle document is not probative of what was known to one of ordinary skill in the art at the time of the invention. More importantly, as discussed above, we find the written description of the '808 patent does not support a limiting construction of the term "database block" based on the purported "application level" versus "storage level" distinction. Further, we do not find any disclosure in the Specification that describes or discusses whether a "database block" is an "application level" entity or construct. Hence, we find the testimony of Dr. Shenoy and the external documents discussed by Patent Owner regarding the "application level" versus "storage level" distinction to be largely divorced from the written description, and, therefore, insufficient to overcome the plain language of the claims. *See Kara Tech.*, 582 F.3d at 1348; *Phillips*, 415 F.3d at 1318.

[14] The page numbers for Exhibit 2102 refer to the page numbers inserted by Patent Owner at the bottom of each page.

3. Database Blocks, Blocks of Data, and Data Blocks

Citing the testimony of Dr. Zadok, Petitioner asserts that, while the ’808 patent never mentions the term “application level,” the “database blocks” recited in the challenged claims in fact encompass storage level “data blocks,” contrary to Patent Owner’s contention. Pet. Reply 7–8 (citing Ex. 1166 ¶ 8). In support of its argument, Petitioner cites the passages in the ’808 patent showing the patent uses “database block” interchangeably with a “block of data.” *Id.* at 11 (citing Ex. 1101, col. 10, ll. 13–23). Citing the testimony of Dr. Zadok, Petitioner asserts that a “block of data” is a broad term that refers to any block, including a storage level disk block. *Id.* (citing Ex. 1166 ¶ 26). Petitioner also points to Patent Owner’s patents that “repeatedly use [the terms ‘database blocks’ and ‘data blocks’] interchangeably.” *Id.* at 7–8 (citing Ex. 1142, col. 15, l. 66–col. 16, l. 18; Ex. 1158 ¶ 52).[15]

Patent Owner asserts that the ’808 patent Specification does not use the terms “database blocks” and “blocks”—that is, “data blocks” or “blocks of data”—interchangeably. *See* PO Resp. 19. Patent Owner argues that the ’808 patent uses “blocks” in the context of “blocks of virtual database files that *point* to database blocks” or uses the term “database block” and “then uses the word ‘block’ as shorthand to refer to these database blocks.” *Id.*

The record supports Petitioner and does not support Patent Owner. Patent Owner’s argument does not account for the complete disclosure of the ’808 patent, which provides multiple examples of using “blocks” or “blocks

[15] Exhibit 1142 is U.S. Patent No. 8,468,174 B1, which is the subject of a related proceeding IPR2015-00128.

of data" interchangeably with "database blocks." For example, the '808 patent states "[t]he *database blocks* retrieved by a point in time copy manager 310 . . . can be used to reconstruct a copy of a database in the production system 110." Ex. 1101, col. 10, ll. 6–10 (emphasis added). In the very next paragraph, the '808 patent continues this discussion by stating that "the point-in-time copy manager 310 may call APIs of storage allocation manager to save *blocks of data* retrieved from the production database system 110" and that "[t]he storage allocation manager 365 keeps track of the various versions of each *block of data* that may be obtained from the production database system 110." *Id.* at col. 10, ll. 14–19 (emphases added). The rest of the paragraph describes managing point-in-time copies of "blocks of data" obtained from the production database system as well as copying, reading, and writing to the "blocks of data." *Id.* at col. 10, ll. 19–30. The disclosure in this paragraph parallels the disclosure in another part of the '808 patent Specification that describes saving and managing of "database blocks" by point-in-time copy manager 310 and storage allocation manager 365. *See id.* at col. 14, ll. 32–44 ("The point-in-time copy manager 310 sends 435 a request to the storage allocation manager 365 to save 535 the *database block*. . . . The storage allocation manager 365 may keep several different versions of the *database block* in the storage system data store 390 . . . if it is updated at different points in time.") (emphases added). Hence, the '808 patent uses the terms "block of data" and "database blocks" interchangeably.

Patent Owner admits that "'data blocks' refer to *blocks of data* stored on a disk and manipulated by the file system." PO Resp. 24 n.7 (emphasis added). Therefore, the ’808 patent Specification does not distinguish between data blocks, blocks of data, and database blocks. This implies that the terms have similar meanings in the context of the ’808 patent. Based on the foregoing, we are persuaded that the “database blocks” recited in the challenged claims do not exclude storage level data blocks, and, therefore, need not include metadata.

4. Whether a Database Block Must Have a Specific Number of Bytes

In the Institution Decision, we found that database blocks of the ’808 patent need not store metadata or have a specific number of bytes stored in storage because an empty database block would not have metadata or “a specific number of bytes stored in storage.” Inst. Dec. 9. Citing the testimony of Dr. Shenoy and the passages in the ’808 patent describing the streaming embodiment, Patent Owner continues to argue that this is a requirement of the proper construction of “database blocks.” PO Resp. 18–19 (citing Ex. 1101, col. 13, ll. 43–50, 53–56; Ex. 2111 ¶¶ 77, 79, 80).

Patent Owner’s argument is unpersuasive because, as discussed above, the streaming embodiment cited by Patent Owner does not limit the claims to require metadata in database blocks. Furthermore, Patent Owner does not explain how to interpret “a specific number of bytes stored in storage.” For example, it is not clear if “a specific number” is a constant. In another argument, Patent Owner states that “[d]atabase blocks may be implemented as *any number of different sizes* independently of the size of the file system data blocks which may ultimately store the database data.” *Id.* at 26 (emphasis added) (citing Ex. 2111 ¶¶ 122–35). Patent Owner also states that "an IBM DB2 database page can be up to 32KB in size, spanning 8 WAFL data blocks, most of which will not contain any metadata for the database page." *Id.* at 25 (citing Ex. 2111 ¶ 129). Hence, according to Patent Owner's arguments, a database block may be of "any number of different sizes." *Id.* at 26. Therefore, database blocks of the '808 patent need not store metadata or have a constant specific number of bytes stored in storage.

5. "Database Block" vs. "File" Distinction

Next, Patent Owner argues that the construction proposed by Petitioner (Pet. 10) and preliminarily adopted in our Institution Decision (Inst. Dec. 11–12)—"a unit of data used by a database"—is improper because it would equate a "database block" with a "file." PO Resp. 22–23. Citing dictionaries, Patent Owner argues that the ordinary meaning of "file" is "a unit of data." *Id.* at 22–23 (citing Ex. 2113, 3; Ex. 2114, 3). Patent Owner further asserts that our preliminary construction incorrectly encompasses a log file described in the Specification because a log file is also "a unit of data used by database." *Id.*

Petitioner argues, citing the deposition testimony of Patent Owner's expert, Dr. Shenoy, Patent Owner is incorrect because a file has a name associated with it but a database block, as construed by Petitioner, would not. Pet. Reply 13 (citing Ex. 1133, 261:19–20, 262:8–9). We agree with Petitioner that defining a database block as "a unit of data used by a database" does not equate a database block with a file.

First, a file is not necessarily "a unit of data used by a database" because not all files are used by a database. Conversely, the phrase "a unit of data used by a database" is not sufficient to describe a file because a file has additional properties or characteristics, such as a unique file name. In fact, the full definitions of a "file" provided in the Patent Owner cited dictionaries, including the portions Patent Owner had omitted when quoting the sources in the Patent Owner Response, show that a file name is a required attribute of a file. *See* Ex. 2113, 3 (defining "file" as "[a] document or other collection of information stored on a disk and identified as a unit **by a unique name**"), Ex. 2114, 3 (defining "file" as "[a] collection of data or information that is stored as a unit in the computer under a single name, **called the *file name***") (the portions omitted by Patent Owner indicated with bold emphasis). Hence, the fact that a definition of a "database block" reads on some "files," such as a log file used by a database, does not mean the definition equates the two terms because a file has additional properties or characteristics, such as a file name.

6. Conclusion

In summary, we find nothing in the intrinsic record, including the written description and the language of the claims, that justifies limiting the term "database block" by reading in the "metadata" or "specific number of bytes" limitations not found in the claims. Furthermore, no extrinsic evidence, including the testimony of experts, presented in this case is sufficient to overcome the plain claim language of the term "database block." Therefore, on this record, consistent with the term as defined in the Specification, we construe "database block" to mean "a unit of data used by a database." *See Phillips*, 415 F.3d at 1316 ("The construction that stays true to the claim language and most naturally aligns with the patent's description of the invention will be, in the end, the correct construction.") (citation omitted).

B. Virtual Database

In the Institution Decision, we preliminarily construed the term "virtual database" to mean "a set of database files capable of being read from and written to, and capable of being mapped to physical addresses for stored database blocks." Inst. Dec. 17. Petitioner agrees with this construction. Pet. Reply 13. Patent Owner, on the other hand, asserts that this definition reads out the "virtual" requirement. PO Resp. 29–30. Patent Owner proposes that the term, instead, should be interpreted to mean "a set of files to which a database server can read and write such that the physical implementation of the database files is decoupled from the logical use of the database files by the database server." *Id.* at 27.

To unpack the language of Patent Owner's proposed construction for further analysis, Patent Owner defines a virtual database as "a set of files" and modifies the definition with two phrases, each expressing a concept involving a database server: (1) "a database server can read and write" to the set of files, and (2) referring to the set of files as "database files," "the physical implementation of the database files is decoupled from the logical use of the database files by the database server." We address each proposed modifying phrase in turn.

1. Whether Reading and Writing by "a Database Server" Is Necessary

Patent Owner argues that the point of the first proposed modifying phrase is to "require[] something of the files . . . that they are of a form readable and writable by a database server." PO Resp. 31. In other words, the main thrust of Patent Owner's proposed phrase is that a database server can read from and write to a set of files.

Independent claims 1 and 50 specifically recite "allowing the database server to read from and write to the set of files [associated with the virtual database]," which encompasses essentially the same concept as Patent Owner's proposed phrase. Hence, including the phrase "a database server can read and write [to a set of files]" in the definition of the stand-alone term "virtual database" as an inherent attribute of the term would render the recited limitation superfluous. Such a construction is presumed improper. *See Digital-Vending Servs. Int'l, LLC v. Univ. of Phoenix, Inc.*, 672 F.3d 1270, 1274–75 (Fed. Cir. 2012) (rejecting the district court's construction narrowing a term by a superfluous limitation when the claims explicitly recited the narrowing limitation, and discussing the "well-established rule that claims are interpreted with an eye toward giving effect to all terms in the claim") (quoting *Bicon, Inc. v. Straumann Co.*, 441 F.3d 945, 950 (Fed. Cir. 2006)) (internal quotation marks omitted); *LSI Indus., Inc. v. ImagePoint, Inc.*, 279 F. App'x 964, 972 (Fed. Cir. 2008) (nonprecedential) (rejecting the district court's construction of "display device" as necessarily including the superfluous limitation of "internal illumination" because other claim terms specifically recited an "illuminated display device"); *but cf.*

ERBE Elektromedizin GmbH v. Canady Tech. LLC, 629 F.3d 1278, 1286 (Fed. Cir. 2010) ("no canon of [claim] construction is absolute in its application") (citation omitted).

Patent Owner also argues that requiring a "database server" to be able to read "a set of files" is necessary to clarify that "a set of files" in its proposed definition must be "database files." PO Resp. 31. Patent Owner's argument is unpersuasive because our construction already requires a virtual database to include "a set of database files." The '808 patent tracks our construction and describes that a virtual database comprises "database files." *See, e.g.*, Ex. 1101, col. 5, ll. 18–21 ("The virtual databases are 'virtual' in the sense that the physical implementation of the *database files* is decoupled from the logical use of the *database files* by a database server."), 39–40 ("A virtual database may be created on a database server by creating the *database files* . . . .") (emphases added).

Patent Owner further argues that "[t]o be a database file, a database server must be able to read, write and understand its contents." PO Resp. 31 (citing Ex. 2111 ¶ 96). As noted above, claim 1 specifies that a database server can read from and write to the set of files for a virtual database. To the extent Patent Owner is arguing *only* database servers can use database files (*see id.* 41 ("*only* the database server has knowledge of and manipulates the database block") (emphasis added); Ex. 2111 ¶ 125 ("The database blocks [are] used *only* by the database server at the application level.") (emphasis added)), Patent Owner's argument does not comport with the written description of the '808 patent. The '808 patent provides a description of a "database" as follows: "[a] database comprises data stored in a computer for use by *computer implemented applications*." Ex. 1101, col. 4, l. 67–col. 5, l. 1 (emphasis added). During the hearing, Patent Owner clarified that that a database server in its proposed construction "is used synonymously . . . with database application." Tr. 184:1–185:12. In line with Patent Owner's acknowledgement at the oral hearing, the '808 patent, and the meaning of a database, a database file may be used by any application or program, not just a "database server." To the extent Patent Owner argues "database files" are limited to data files used in relational database systems, such as the Oracle database system, that can be understood *only* by a relational database server, such as the Oracle database server, *see, e.g.*, Ex. 2111 ¶ 125 (quoted above) (citing an Oracle technical document (Ex. 2101) describing the features of "the Oracle database server, an object-*relational* database management system" (Ex. 2101, 25) (emphasis added)),[16] Patent Owner's argument is unpersuasive because "database" recited in the challenged claims is not limited to a relational database for the reasons discussed above in Section III.A.1.d.

The Specification makes clear that the readable and writable characteristics are inherent attributes of a virtual database. *See, e.g.*, Ex. 1101, col. 18, ll. 27–28 ("FIG. 10 indicates how storage efficient copies are made to create a *read/write file structure representing a VDB*."); col. 17, ll. 57–63 ("The virtual database manager [] identifies [] the recent most PIT copy associated with time Tj . . . The *read/write file structure* [] is created [] by making storage efficient copies of the database blocks in the identified

[16] The page numbers for Exhibits 2101 refer to the page numbers inserted by Patent Owner at the bottom of each page.

PIT copy."); col. 18, ll. 12–15 ("The virtual database manager [] sends . . . handles to the *read/write file structure* to the associated virtual database system 130.") (emphases added). Thus, a virtual database must comprise readable and writable database files. However, the phrase "database files" need not be further modified to indicate that the files are intended for use by a computer program because the concept is already included in the description of "database" provided in the Specification, as discussed above. Therefore, the term "virtual database" is properly construed, to have as the base phrase of its definition, "a set of readable and writable database files."

2. The Meaning of the Phrase "the Physical Implementation of the Database Files is Decoupled from the Logical Use of the Database Files by the Database Server"

Patent Owner relies on the following sentence in the '808 patent to support the second proposed modifying phrase, i.e., the "decoupled" aspect, of its proposed construction: "virtual databases are 'virtual' in the sense that the physical implementation of the database files is decoupled from the logical use of the database files by a database server." PO Resp. 28 (quoting Ex. 1101, col. 5, ll. 18–21). Patent Owner asserts this sentence explicitly defines the use of the word "virtual" in the term "virtual database." *Id.* However, the '808 patent does not describe expressly what it means for "the physical implementation of the database files [to be] decoupled from the logical use of the database files by a database server." Citing the testimony of Dr. Shenoy, Patent Owner asserts the cited sentence describes one of the key features of the invention of the '808 patent, which is applying the general concept of virtualization—"the *decoupling* of a computing process *from the platform* on which it operates"—to databases. *Id.* (emphases added) (citing Ex. 2111 ¶¶ 23–26, 63–70, 87–90). Petitioner asserts Patent Owner confuses "platform virtualization" with "database virtualization" or "virtualizing data." *See* Pet. Reply 15–16. We agree with Petitioner that "platform virtualization" is unrelated to the concept of "virtual database" as claimed and described in the '808 patent.

In his Declaration, Dr. Shenoy testifies that the Java programming language running on a Java Virtual Machine as described in an Oracle document explains the general concept of platform virtualization. *See* Ex. 2111 ¶¶ 23, 24 (citing Ex. 2101, 451–452, Fig. 24-7). Patent Owner's discussion of the Java Virtual Machine illustrates the basic problem with Patent Owner's argument—that it is directed to the features or functions of software, not a database. For example, Dr. Shenoy explains that the "decoupling [of] a computing process from the platform on which it operates" is achieved by the Java Virtual Machine, which "is *software* that emulates the operation and interface of a physical processor." *See* Ex. 2111 ¶¶ 23–24 (emphasis added). The '808 patent also describes that a "virtual machine" is "provided by *platform virtualization software*" or "server virtualization *software*." Ex. 1101, col. 7, ll. 16–19 (emphases added). Further, the '808 patent describes a Java Virtual Machine as part of production database system 110 (Ex. 1101, col. 8, ll. 12–14), which is separate from virtual database 220 stored on database storage system 100, as discussed above in Section II.A. Hence, Patent Owner's discussion of the Java Virtual Machine may relate to database servers or other *software* of the disclosed system, but it is *not* directed to a virtual database, which is a set of *files* as expressly recited in the claims and described in the Specification (*see, e.g.*, Ex. 1101, col. 6, ll. 49–51; Fig. 2a) and, therefore, is not persuasive even as an analogy.

Other examples of "decoupling" Dr. Shenoy discusses similarly relate to the function or operation of various software or programs, not database files. For instance, Dr. Shenoy testifies that, because the system of the '808 patent operates at the "application level" using APIs, the database storage system of the invention does not depend on and need not concern itself with the underlying details of the physical storage system. Ex. 2111 ¶¶ 69, 70 (quoting Ex. 1101, col. 7, ll. 47–52). The cited portion of the '808 patent describes, however, that the APIs are provided by "production system library 385" and "vendor interface module 335," which are programs or software code modules residing at the production system. *See, e.g.*, Ex. 1101, col. 7, ll. 52–55 ("An example of a vendor interface module is the *program code* of a database server provided by vendor ORACLE that implements RMAN APIs."), col. 13, ll. 10–12 ("the production system library 385 includes *code* to analyze the structures of the files of the database stored in the data store 350") (emphases added). In another instance, Patent Owner cites the operation of file sharing manager 370, which is a program or software module running at database storage system 100 (*see* Ex. 1101, col. 9, ll. 38–42; col. 10, ll. 32–54; Fig. 3), as an example that "[t]he physical implementation of the files that are created to implement the virtual database is thereby decoupled from the logical use of the database files by a database server." *See* PO Resp. 15–16 (citing Ex. 1101, col. 10, ll. 50–54; Ex. 2111 ¶¶ 70–71).

In addition, Patent Owner relies upon the testimony of Dr. Shenoy that "each database (130c)" depicted in Figure 1 of the '808 patent is "virtual" (Ex. 2111 ¶ 89). PO Resp. 28 (citing Ex. 2111 ¶ 89)). However, Figure 1 shows 130c as the "Virtual Database System 130(c)," instead of the disclosed virtual database, which is stored in a different system, namely, Database Storage System 100. *See* Ex. 1101, Figs. 1, 2a, 2b. That is, virtual database *system* 130 is separate from and external to *virtual database* 220. *Id*. Further, the '808 patent describes that virtual database system 130 includes no database or database files but, rather, comprises a database server and a VDB system library, both of which are software modules or programs. *See* Ex. 1101, col. 12, ll. 3–13 ("A virtual database system 130 includes a database server 360 and a VDB system library 380. . . . The VDB system library 380 contains *program code* for processing requests sent by the database storage system 100.") (emphasis added); Fig. 3. Hence, Patent Owner appears to conflate the function of the *software* of a virtual database *system* with the meaning of "a virtual database," which is a set of *files* that resides in a database storage system that is separate from the virtual database system.

Petitioner argues that Patent Owner's platform virtualization argument is misguided because it conflates the meaning of a "virtual database" with the disclosed embodiments of production database *system* and the virtual database *system*. Pet. Reply 16. We agree with Petitioner as discussed above. Patent Owner's arguments are directed to the software part of the disclosed system that is separate from and external to a virtual database. As described in the '808 patent, a database system generally comprises a database, which is data stored in storage, and database software, such as a database servers or other program that accesses the database. *See* Ex. 1101, col. 4, l. 67–col. 5, l. 4; col. 7, ll. 35–57. Dr. Shenoy similarly describes a database system as comprising a database and database software:

> A *database system* is typically a computer that includes the stored *data itself*, i.e. *the database*, as well as *database management software*, often shortened to DBMS. A DBMS includes *software* for accessing the data in the database and modifying that data to add, remove or change stored data. A DBMS may run as or include an autonomous *program* which handles client requests to the database and which is often called a *database server*.

Ex. 2111 ¶ 34 (emphases added). Hence, the platform virtualization or the storage virtualization or abstraction argued by Patent Owner as demonstrating what is meant by the phrase "the physical implementation of the database files is decoupled from the logical use of the database files by a database server" is a function or feature of the *software part* of a database system—be it Java Virtual Machine, a database server, or various software modules providing APIs—which is separate from and external to a database, i.e., the *data part* of a database system. Therefore, contrary to Patent Owner's argument, virtualization of a computing platform or storage by software is unrelated to and cannot be equated to virtualization of data, i.e., the subject matter claimed by the term "virtual database."

In effect, Patent Owner's platform virtualization argument calls for improperly injecting the functional features of the software of the disclosed system into the meaning of a virtual database, which is a set of files separate from and external to the software part. A proper definition of the term "virtual database" must focus on the characteristics of the database file itself, e.g., the structure of the database file, not the function of any particular software embodied in the program instructions of the software program that accesses the database file. If the patentee intended to claim the function or the operation of the software of the disclosed system, it could have done so by explicitly claiming the subject matter. Not having done so, Patent Owner may not import the array of functions and intelligence embodied in the described software into a single term "virtual database" to impart a very different meaning to the claims from what is indicated by the plain language of the claims and the written description of the patent. Patent Owner asserts that, by discounting Patent Owner's platform virtualization argument, the preliminary construction reads out the "virtual" requirement from the term "virtual database." PO Resp. 29–30 (citing Inst. Dec. 17). Patent Owner's argument is unpersuasive because our refusal to read the functional features of the software modules into the meaning of "virtual database" does not equate reading out the word "virtual" from the term. On the contrary, as discussed below, the preliminary construction is consistent with the language of the claims and the written description of the '808 patent.

Turning to the claim language and the written description pertaining to the structure of virtual database files, the challenged independent claims recite, with emphasis added, creating "a set of files for a virtual database, each file in the set of files [is] *linked* to the database blocks on the storage system associated with a point-in-time copy of the source database." Tracking the language of the claims, the Specification describes as follows:

> A set of files are created for a virtual database. Each file in the set of files created for a VDB is *linked* to the database blocks on the storage system associated with a point-in-time copy of the source database.

Ex. 1101, col. 2, ll. 24–27 (emphasis added); *see also id.*, Abstract (making essentially the same statement). Hence, the database files of a virtual database are linked or mapped to the database blocks associated with the database being virtualized.

Looking to the Specification for further clarification, Figure 10 of the ’808 patent depicts “how . . . to create a read/write file structure representing a VDB.” *Id.* at col. 18, ll. 27–28. When the VDB file structures are created, “the blocks V11, V12, . . . , V25 may be implemented as *pointers* to the actual database block that stores the data.” *Id.* at col. 18, ll. 47–49 (emphasis added). The ’808 patent further describes:

> For example, V11 represents the information in block F1 and since the block F1 was never updated during copies made at time T1 and T2, *V11 points at F11*. V12 represents the information in block F2 and since F2 was updated at time T1, *V12 points at the block F22*. Similarly, *V13 corresponds to block F3* that was updated at time T2 and *points at the block F33*.

Id. at col. 18, ll. 49–55 (emphases added). Hence, blocks V11, V12, etc. of a virtual database do not contain ordinary data, but, instead, point to the location or address of the actual database blocks stored in a physical storage device. Another example implies that writing to a pointing block in a database file, such as V11, which is stored at one location, actually results in writing to another physical block location such as F11: “For example, if the virtual database system 130 writes to the block V11, space is allocated and block F11 copied to the allocated block." *Id.* at col. 19, ll. 51–53. The '808 patent further describes "[s]ince the [virtual database file] structure 1050 illustrated in FIG. 10, structure 1150 illustrated in FIG. 11, or structure 1350 illustrated in FIG. 13 are read/write structures, the virtual database system 130 is allowed to read from these structures as well as write to them." *Id.* at col. 19, ll. 44–48. Thus, the '808 patent indicates that the database files are "virtual" in the sense that they create the illusion of allowing data to be written to pointing blocks in the virtual database files, but the system actually writes the data elsewhere to another physical location specified by the pointing blocks such as V11 and V12. *See id.* at col. 19, ll. 44–53, Figs. 10–12.

Based on the disclosure in the '808 patent describing the structure of virtual database files, "the logical use of database files" can be understood as reading from or writing to database blocks by reading from or writing to virtual database file structures that point to the actual database blocks. Hence, as described in the '808 patent, decoupling of physical implementation of the database files, i.e., actual database blocks stored on a physical storage device, from the logical use of the database files, i.e., accessing the database blocks through virtual database files, is accomplished by using pointers to map blocks in virtual database files to physical addresses for database blocks stored in physical storage devices.

In fact, although not discussed in the Patent Owner Response, Dr. Shenoy acknowledges that Figure 10 illustrates the "decoupling of the physical storage of the database from its logical use" (Ex. 2111 ¶ 63) and that "the physical storage for the virtual database is decoupled from the logical view of the database . . . by *mapping* blocks captured at various points in time to [virtual database files]" (*id.* ¶ 94) (emphasis added).

Nonetheless, Patent Owner argues that the preliminary construction is erroneous because database files of any database, including a non-virtual database, such as a source database, can be "mapped to physical addresses for stored database blocks." PO Resp. 30 (citing Ex. 2111 ¶ 97).

Patent Owner's argument is partially persuasive.[17] Accordingly, we modify our preliminary construction. To the extent that Patent Owner is arguing the preliminary construction does not distinguish a non-virtual source database from a virtual database, the concern can be addressed by expressly indicating that the stored database blocks are associated with *another* database. As Patent Owner's arguments imply, an ordinary (i.e., a non-virtual) database consists of its own database blocks and would *not* be created by mapping or pointing to stored blocks associated with *another* database. Therefore, based on the foregoing discussion, the term "virtual database" is construed as "a set of readable and writable database files capable of being mapped to physical addresses for stored database blocks associated with another database." The claim construction tracks the language of the challenged claims, the written description, Dr. Zadok's testimony, and Dr. Shenoy's testimony, all of which show how to create a

[17] Patent Owner's argument is partially unpersuasive because the '808 patent expressly claims and describes that a source database can be a virtual database. *See* Ex. 1101, col. 2, ll. 30–32; col. 19, ll. 17–18; claim 6.

virtual database (i.e., using a set of files), as discussed above. *See* Ex. 2111 ¶ 95; Ex. 1101, Abstract.

IV. PRINTED PUBLICATION

Patent Owner in its Response contests that Sanders is a prior art "printed publication" in accordance with 35 U.S.C. §§ 102 and 311(b). PO Resp. 1–4. We look to the underlying facts to make a legal determination as to whether a document is a printed publication. *Suffolk Techs., LLC v. AOL Inc.,* 752 F.3d 1358, 1364 (Fed. Cir. 2014). The determination of whether a document is a "printed publication" under 35 U.S.C. § 102(b) involves a case-by-case inquiry into the facts and circumstances surrounding its disclosure to members of the public. *In re Klopfenstein,* 380 F.3d 1345, 1350 (Fed. Cir. 2004). A key inquiry is whether the reference was made "sufficiently accessible to the public interested in the art" before the critical date. *In re Lister*, 583 F.3d 1307, 1311 (Fed. Cir. 2009) (quoting *In re Cronyn*, 890 F.2d 1158, 1160 (Fed. Cir. 1989)).

Initially, we note our disagreement with Patent Owner's contention that Petitioner cannot rely upon evidence not submitted with the Petition to show that Sanders is prior art. PO Resp. 2–4. In Patent Owner's view, Petitioner must make out a *prima facie* case of unpatentability in its Petition, which includes the substantive element of Sanders being publicly accessible and prior art. *Id.* at 3–4. That position, however, does not account for the difference between the threshold for instituting a trial (35 U.S.C. § 314(a)) and proving unpatentability of a claim in trial (35 U.S.C. § 316(e)). As noted by our reviewing court, "there is a significant difference between a petitioner's burden to establish a 'reasonable likelihood of success' at institution, and actually proving invalidity by a preponderance of the evidence at trial." *TriVascular, Inc. v. Samuels*, 812 F.3d 1056, 1068 (Fed. Cir. 2016) (quoting 35 U.S.C. § 314(a) and comparing § 316(e)).

Based on the information presented in the Petition and Patent Owner's Preliminary Response, we determined there was a reasonable likelihood that Petitioner would prevail in its challenges that included Sanders. Inst. Dec. 45; *see* 35 U.S.C. § 314(a) (threshold for instituting *inter partes* review); *see also* 37 C.F.R. § 42.108(c) ("The Board's decision [on Institution] will take into account a patent owner preliminary response where such a response is filed."). Patent Owner did not challenge the prior art status of any of the applied patents or publications in its Preliminary Response. Patent Owner, in fact, stated that it had "disclosed to the Patent Office every NetApp feature that Petitioner now cites in the Petition" and that the NetApp references including Sanders "were *published* over a span of six years." Prelim. Resp. 50 (emphasis added). We do not mean to suggest that a patent owner must raise any "printed publication" issues in a preliminary response in order for the Board to consider such issues in the preliminary proceeding phase. In this case, however, based in part on the information in Patent Owner's Preliminary Response and in part on the printed dates and the lack of indicia of confidentiality or internal, non-public distribution in Sanders, we determined that Petitioner had met its burden for a threshold showing to proceed to trial.

Patent Owner also argues that Petitioner cannot rely on declarations filed after the Petition. These declarations include two declarations provided by Louis Hernandez (Ex. 1122) and Joseph Ortiz (Ex. 1130) in response to objections by Patent Owner[18] and a Supplemental Declaration by Mr. Hernandez (Ex. 1150) filed with its Reply. PO Resp. 2–4. Our rules authorize serving supplemental evidence in response to an objection. 37 C.F.R. § 42.64(b)(2). Patent Owner lacks a basis to complain that evidence has been produced in response to its objections. Petitioner also relies, properly, on the supplemental evidence in its Reply, as evidence in reply to Patent Owner's arguments in its Response that Sanders is not a printed publication.

Turning to the substance of Exhibit 1122, Mr. Hernandez testifies that he is currently employed by Petitioner, was employed by NetApp from 2004 to 2009, and was a NetApp customer from 2000 to 2004. Ex. 1122 ¶¶ 1, 2, 4. Mr. Hernandez testifies further that for most of his time at NetApp, as a Systems Engineer, he was responsible for marketing NetApp's products and services to numerous customers, prospective customers, business partners, and/or alliances. *Id.* ¶ 3. Mr. Hernandez further testifies: "During the 2000-2009 time-frame, to support its marketing efforts, it was NetApp's standard practice to publish technical reports, white papers, and product manuals or guides to customers, potential customers, business partners, and alliances." *Id.* ¶ 6. He also testifies: "These documents were published, according to standard practice, as of the month and year that appeared on the face of the documents." *Id.* Mr. Hernandez testifies that he has personal knowledge of and recognizes Sanders, and that it was published during his tenure at

[18] Exhibit 1130 is expunged at Petitioner's request. We do not further discuss the Exhibit.

NetApp or his subsequent tenure at Midwave, a certified distributor and reseller of NetApp. *Id.* ¶¶ 7, 11, 14, 17.

Patent Owner argues Mr. Hernandez does not declare that Sanders was "publicly accessible." PO Resp. 2. Patent Owner submits as follows:

> Even if it was NetApp's "standard practice" to provide its documents to its "customers, potential customers, business partners and alliances," that does not establish that these documents were available to *the public*, but instead shows at most that they were only available to a subset of entities affiliated with NetApp.

Id. at 2–3.

Petitioner replies with a Supplemental Declaration from Mr. Hernandez. Pet. Reply 6–7 (citing Ex. 1150 ¶¶ 3–13). Mr. Hernandez testifies that he uses the term "publish" or "published" as referring to documents being publicly distributed to customers, potential customers, business partners, and alliances as of the month and year that appeared on the face of the documents, non-confidentially. Ex. 1150 ¶ 5. According to Declarant, NetApp had more than two hundred Systems Engineers and other sales personnel during the relevant timeframe (*id.* ¶ 4) and that technical reports, white papers, product manuals, and product guides were freely distributed to support its marketing efforts (*id.* ¶ 7). Mr. Hernandez testifies further that it was important for NetApp to date the documents accurately so that customers and potential customers could understand if a specific document accurately reflects features for specific versions of NetApp's products or if a document was outdated or updated to reflect more current features. *Id.* ¶ 10.

As part of routine discovery (37 C.F.R. § 42.51(b)(1)(ii)), Patent Owner had the opportunity to cross-examine Mr. Hernandez during Patent Owner's first discovery period but elected not to. Patent Owner cross-examined Mr. Hernandez in its second discovery period regarding the testimony in his Supplemental Declaration. We have considered Patent Owner's Motion for Observation on Cross-Examination Testimony of Mr. Hernandez (Paper 47) and Petitioner's Response (Paper 55), insofar as they relate to public accessibility of Sanders.[19] We acknowledge the potential for bias in Mr. Hernandez's testimony as a present employee of Petitioner. We find, however, the testimony in Mr. Hernandez's Declarations as to public accessibility of Sanders to be credible. As an earlier panel of the Board has found, in a proceeding involving a different patent and different parties, documents such as Sanders are dated technical documents or whitepapers, having no indication of being mere drafts or internal papers, each of which is "a type of document whose very purpose is public disclosure." *Veeam Sw. Corp. v. Symantec Corp.*, Case IPR2014-00089, slip op. at 14 (PTAB Apr. 25, 2014) (Paper 9). Moreover, in that case, at the preliminary proceeding phase, the panel found that Sanders, on its face, was sufficient to qualify as a printed publication. *Id.* at 2, 13–14.

Finally, Petitioner also submits a declaration it says was produced in response to Patent Owner's evidentiary objections. Pet. Reply 4. Petitioner

[19] During the Hearing, Patent Owner asked for, and we granted, additional time to consider its oral Hearing arguments regarding alleged new issues (regarding publication) raised in Petitioner's Reply in lieu of filing a Sur-Reply. *See* Tr. 211:16–212:18; 224:13–21; 237:1–25.

provided the declaration from the office manager of the Internet Archive (Wayback Machine). *Id.* (citing Ex. 1125). Sanders is dated April 30, 2006 and is designated "TR-3460" (or Technical Report 3460). Ex. 1105, 1. According to the testimony regarding how the Internet Archive works (Ex. 1125 ¶¶ 3–5), we find the evidence indicates that Sanders was available on NetApp's commercial website on, or at least as early as, November 22, 2006. *Id.* at 231, 232.[20] Exhibit 1125 indicates that Sanders was, thus, "sufficiently accessible to the public interested in the art." *See Lister*, 583 F.3d at 1311.

"A given reference is 'publicly accessible' upon a satisfactory showing that such document has been disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art exercising reasonable diligence, can locate it." *SRI Int'l, Inc. v. Internet Sec. Sys., Inc.*, 511 F.3d 1186, 1194 (Fed. Cir. 2008) (quoting *Bruckelmyer v. Ground Heaters, Inc.*, 445 F.3d 1374, 1378 (Fed. Cir. 2006)). In view of the foregoing considerations, we find that Petitioner has established, by a preponderance of the evidence, that Sanders (dated Apr. 30, 2006) was available to the interested public at least more than one year before October 21, 2009, the earliest possible priority date of the '808 patent. *See* Ex. 1101, at (22). Therefore, on this record, Sanders is a printed publication under 35 U.S.C. § 102(b).

[20] The page numbers for Exhibit 1125 refer to the page numbers inserted by Petitioner in the bottom, right-hand corner of each page.

V. ANALYSIS OF PETITIONER'S PRIOR ART CHALLENGES

A. Obviousness Based on the Combination of Edwards, Patterson, and Sanders

In the '016 IPR, Petitioner asserts claims 3, 29, 31, 53, 54, and 56 are unpatentable under 35 U.S.C. § 103(a) over the combination of Edwards, Patterson, and Sanders. Pet. 36–54. In the '019 IPR, Petitioner asserts that the same combination of prior art renders claims 2, 24–26, and 51 obvious under § 103(a). '019 Pet. 32–52.

Claims 2, 3, 24–26, 29, and 31 depend from claim 1, and claims 51, 53, 54, and 56 depend from claim 50. Hence, in our discussion below, we consider independent claims 1 and 50 first before addressing the challenged dependent claims. With respect to common base claims 1 and 50, the parties' unpatentability arguments advanced in the '016 IPR are essentially identical to their arguments presented in the '019 IPR. Hence, although our discussion of claims 1 and 50 below focuses on the '016 IPR, our findings and conclusions apply equally to the '016 IPR and the '019 IPR cases unless indicated otherwise.

Upon review of all of the parties' papers and supporting evidence discussed in those papers, we are persuaded that Petitioner has demonstrated, by a preponderance of evidence, that claims 2, 3, 24–26, 29, 31, 51, 53, 54, and 56 are unpatentable under 35 U.S.C. § 103(a) over the combination of Edwards, Patterson, and Sanders.

1. Principles of Law

A claim is unpatentable under § 103(a) if the differences between the claimed subject matter and the prior art are such that the subject matter, as a whole, would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). The question of obviousness is resolved on the basis of underlying factual determinations, including: (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of skill in the art; and (4) where in evidence, so-called secondary considerations. *Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966). We analyze this asserted ground based on obviousness with the principles identified above in mind.

2. Sanders

Sanders describes a system "to clone a DB2 database quickly and easily." Ex. 1105, 1. "Database cloning is [a] process by which you can create an exact copy of a DB2 database." *Id*. at 3. The disclosed NetApp system uses FlexClone and SnapMirror technologies in a combined manner. *Id*. at 8. This combined technology allows administrators to clone a production FlexVol database system as a writable FlexClone database on another storage system. *Id*. at 8–9. Specifically, FlexClone provides point-in-time copies of the production database. *Id.* at 3. Further, "[a] FlexClone volume is a writable point-in-time image of a FlexVol volume or another FlexClone volume." *Id*. Stated differently, "[t]he clone database is a frozen image of the database file system at the time of the clone creation. If necessary, the primary database can be restored from the snaphsot created for the clone; or applications can point directly to the clone database." Ex. 1105, 6.

A FlexClone volume "uses space very efficiently, allowing both the original FlexVol volume and the FlexClone volume to share common data, storing only the data that changes between the original volume and the clone." *Id.* at 3. Clones can be created on the same or different storage systems. *Id.* at 6. The SnapMirror technology provides FlexClone volumes to be produced at different destinations: "A SnapMirror source and its corresponding destination can reside on the same storage system or on two separate storage systems that are miles apart." *Id.* at 3. Figure 3 of Sanders is reproduced below:

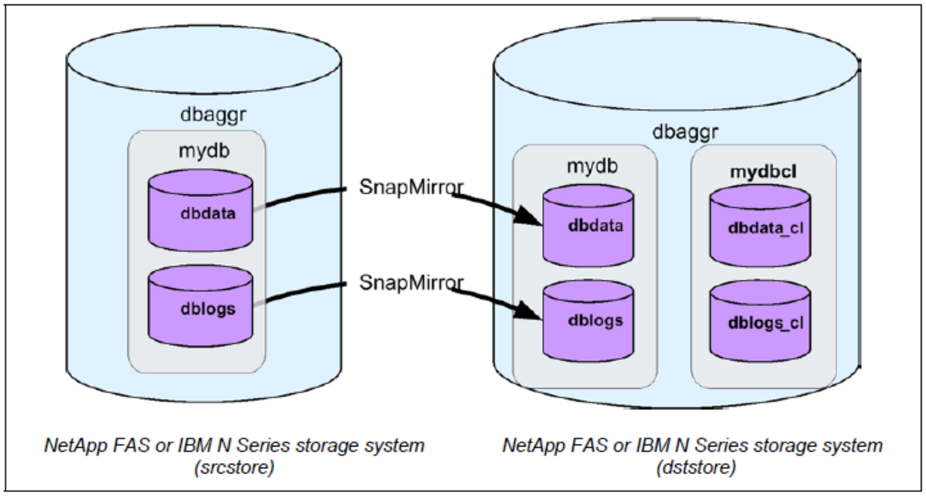

dbaggr
mydb
dbdata
dblogs
SnapMirror
SnapMirror
dbaggr
mydb
dbdata
dblogs
mydbcl
dbdata_cl
dblogs_cl
NetApp FAS or IBM N Series storage system (srcstore)
NetApp FAS or IBM N Series storage system (dststore)

Figure 3) Clone database volumes on the second NetApp FAS or IBM N Series storage system

Figure 3 depicts using the combined SnapMirror and FlexClone technology to transmit point-in-time clone copies of a production database to a destination storage system. *See id.* at 8.

Sanders also describes mounting a clone database on a database server. *See id.* at 14 ("In order to access the clone database, you need to mount the clone volumes to a database server. . . . [Y]ou can mount the clone volume by executing the following command on the database server: mount [MountPoint].").

3. Edwards

Edwards describes the same NetApp system as Sanders, and provides additional explanations regarding the virtualization of volumes and files used to clone DB2 databases in Sanders. A volume essentially comprises a file system that points to underlying data storage on storage disks:

> [A] FlexVol volume is a file system created *within* a file on an underlying file system. A hidden file system spans a pool of storage, and we create externally visible volumes inside files on this file system. This introduces a level of indirection, or *virtualization between the logical storage space used by a volume and the physical storage space* provided by the RAID subsystem.

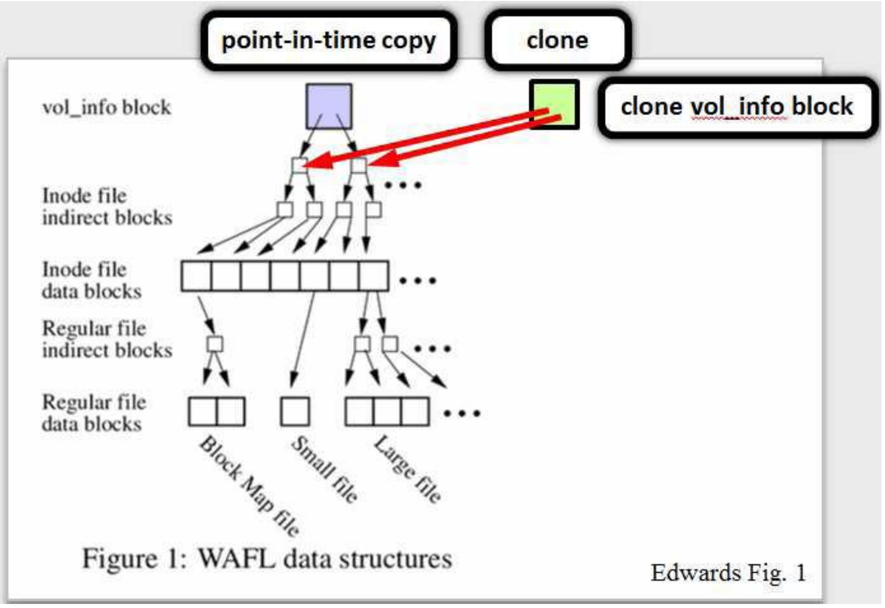
Ex. 1103, 11[21] (second emphasis added). Edwards describes the same SnapMirror and FlexCone systems that Sanders describes, as forming a virtualized system: "[W]e *virtualize* the allocation of volumes on physical storage, allowing multiple, independently managed file volumes, along with their Snapshot copies, to share the same storage." *Id*. at 10 (emphasis added). "*Virtualization* is a well-known method of abstracting physical resources and of *separating* the manipulation and *use of logical resources*

[21] The page numbers for Edwards refer to the page numbers inserted by Petitioner in the bottom, right-hand corner of each page.

from their underlying implementation." *Id.* at 9 (Abstract) (emphases added). "The resulting *virtual file volumes*, or *FlexVol® volumes*, are managed *independent of lower storage layers*." *Id*. (emphasis added). Edwards also describes mapping virtual blocks to the addresses of physical blocks: "*Mapping between virtual block addresses* used by FlexVols and *physical block addresses* . . . ." *Id.* at 10 (emphases added).

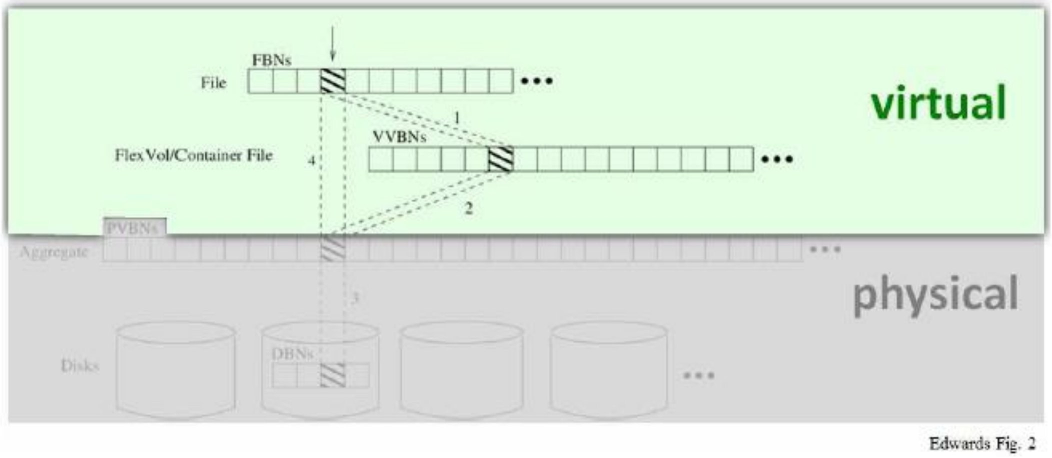
Edwards further explains this virtualization technology is used "to implement writable Snapshot copies (called *FlexClone volumes*)." *Id*. at 10. A clone volume "inherits pointers to the complete file system image stored in the original Snapshot copy" of an original FlexVol volume. *Id*. at 15. The Snapshot copies are point-in-time copies: "The only differences between a Snapshot copy and the live file system are the blocks that have been modified since the Snapshot copy was created (and the metadata that points to them)." *Id*. at 11. "WAFL Snapshot copies provide consistent point-in-time copies of a volume." *Id*. at 14. Although a Snapshot copy is read-only, combining Snapshot and FlexClone technologies provides writable Snapshot copies since "[i]n database environments . . . it is often desirable to make writable copies of a production database for development or test purposes." *Id*.

Edwards explains that "[c]reating a clone volume is a simple process." Ex. 1103, 15. A container file for the new clone volume (or FlexClone volume) is created and seeded "with a vol_info block that is a copy of the vol_info block of the snapshot copy on which the clone is based." *Id*. Because vol_info block is the root of the "tree of blocks that form the snapshot copy, the clone inherits pointers to the complete file system image stored in the original snapshot copy." *Id*. Petitioner provides an annotated version of Edwards's Figure 1, to demonstrate this step in cloning, as follows:



Pet. 20. Figure 1 as annotated by Petitioner depicts WAFL data structures of a Snapshot copy and a clone volume based on the Snapshot copy.

The system uses the above WAFL structures to translate a given file offset in a volume to find a physical block (and to copy blocks), by employing a related system of pointers (VVBN and PVBN). *See* Ex. 1103, 11–13. The file's offset translates to VVBN (virtual volume block number), "a block address within the FlexVol volume's virtual block address space" that "specifies the block's offset within the container file." *Id*. at 11. The VVBN represents a logical address, which points to a PVBN (physical volume block number). *See id.* at 11–12. The PVBN specifies the block's physical location within the aggregate of disks. *Id*. at 11. The RAID subsystem of WAFL ultimately translates the PVBN pointer to a DBN (disk block number), in order to store or retrieve a block. *See id.* at 12, Fig. 2. Petitioner provides an annotated version of Figure 2 of Edwards, to illustrate the "VVBN-to-PVBN mapping," as follows:



Pet. 18. Figure 2 depicts translating or mapping a VVBN to a PVBN using the structures in the container file. *See* Ex. 1103, 12.

Edwards also describes SnapMirror, which mirrors the contents of a volume from a source system to a destination storage system. *Id.* at 13. The VVBN remains constant when SnapMirror transfers data blocks from a source FlexVol volume to a destination FlexVol volume. *Id*. at 14. On the other hand, the destination system, which may be physically different than the source system, may assign a different destination PVBN (i.e., which differs from the source PVBN). *Id*. at 14, Fig. 4. In other words, "flexible volume transfers are VVBN-based." *Id*. at 14. Thus, the "VVBN-to-PVBN mapping at the destination is different from that at the source." *Id.* "This removes geometry restrictions from Volume SnapMirror because the source and destination make physical allocation decision independently. As a result, volumes can be mirrored between aggregates with different sizes and/or disk configurations.” *Id*.

4. Patterson

Patterson describes NetApp’s SnapMirror technology and provides additional explanations regarding creating multiple point-in-time copies using SnapMirror. For example, Figure 1 of Patterson illustrates a situation where two different snapshots of a volume at different times are transferred (mirrored) to a destination storage system. Figure 1 is reproduced below with annotation added.

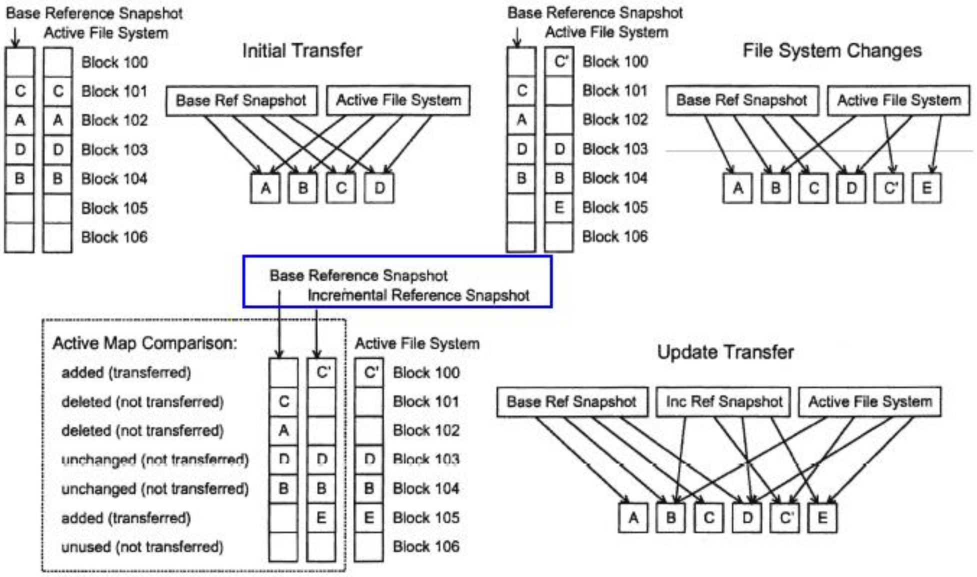


Figure 1. SnapMirror’s use of snapshots to identify blocks for transfer. SnapMirror uses a base reference snapshot as point of comparison on the source and destination filers. The first such snapshot is used for the Initial Transfer. File System Changes cause the base snapshot and the active file system to diverge (C is overwritten with C', A is deleted, E is added). Snapshots and the active file system share unchanged blocks. When it is time for an Update Transfer, SnapMirror takes a new incremental reference snapshot and then compares the snapshot active maps according to the rules in the text to determine which blocks need to be transferred to the destination. After a successful update, SnapMirror deletes the old base snapshot and the incremental becomes the new base.

Figure 1 shows creating Base Reference Snapshot and Incremental Reference Snapshot at different times. In the initial mirror transfer, a base snapshot (Base Reference Snapshot) of the active file system (which represents the source volume's file system in its current state) is taken. *See* Ex. 1104, 11.[22] Both the snapshot and the active file system point to the same data blocks A, B, C, and D. Figure 1 further shows subsequent changes in the active file system, which "cause the base snapshot and the active file system to diverge (C is overwritten with C', A is deleted, E is added)." *Id.* In the next SnapMirror transfer (called "update transfer") from the source volume to the destination storage, "SnapMirror takes a new incremental reference snapshot." *Id.* The incremental reference snapshot is compared with the base reference snapshot to determine which blocks have changed so that only those changed blocks, namely, blocks C' and E, are transferred to the destination. *Id.* Thus, blocks B and D stored in the SnapMirror destination would be associated with at least two point-in-time copies, i.e., Base Reference Snapshot and Incremental Reference Snapshot, of the source file system.

5. Claim 1

a. Whether the Combination of Edwards, Patterson, and Sanders Teaches Every Limitation of Claim 1

(1) Petitioner's Contentions

Petitioner has shown that the combination of Edwards, Patterson, and Sanders teaches every limitation of claim 1. Pet. 36–44. Petitioner points to specific disclosures in the prior art that are deemed to describe or teach all claim limitations. *Id.* In addition, Petitioner relies upon the Declaration of Dr. Zadok (Ex. 1119) to support its positions. *Id.*

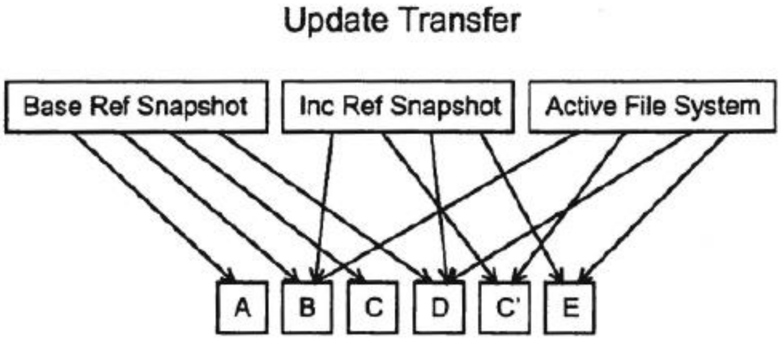
[22] The page numbers for Patterson refer to the page numbers inserted by Petitioner in the bottom, right-hand corner of each page.

Addressing the preamble of claim 1, which recites a method for "creating a virtual database system," Petitioner asserts that it is not limiting because it is duplicative of the limitations in the claim's body and does not recite any essential structure or steps that are not found in the body of claim 1. Pet. 36. Petitioner further argues that the preamble is not necessary to give life, meaning, and vitality to claim 1. *Id.* Patent Owner does not dispute Petitioner's contentions or argue the preamble is limiting. On this record, we find the preamble of claim 1 is not a limitation because the claim body describes a structurally complete invention such that "deletion of the preamble phrase [would not] affect the structure or steps of the claimed invention." *Catalina Mktg. Int'l, Inc. v. Coolsavings.com, Inc.*, 289 F.3d 801, 809 (Fed. Cir. 2002).

The first step of claim 1 recites "receiving different point-in-time copies of a source database, the source database comprising a plurality of database blocks." Petitioner asserts that the Snapshot copies of a WAFL or FlexVol volume described in Edwards teach multiple point-in-time copies of a source volume. Pet. 37–38 (citing Ex. 1103, 10, 11). Petitioner further asserts that Edwards teaches receiving multiple point-in-time copies of a source database because Edwards describes that, using SnapMirror, multiple snapshots of a volume are created at different points in time at a source system and transferred to and received by a destination storage system. *Id.* at 37–38, 39 (citing Ex. 1103, 11, 13, 14, Fig. 4). In addition, Petitioner argues Sanders teaches using a DB2 database as the source database. *See id.* at 31.

Petitioner asserts Edwards also teaches that a source volume comprises a plurality of data blocks (*id.* at 38 (citing Ex. 1103, 10)) and the source volume can include a database. *Id.* at 37 (citing Ex. 1103, 15 ("a 400GB ***database table*** is created on a 1TB FlexVol volume") (emphasis by Petitioner)). Citing the testimony of Dr. Zadok, Petitioner argues these WAFL or FlexVol data blocks are "database blocks" because they are "units of data . . . used by . . . a database." *Id.* at 38 (citing Ex. 1103, 15, 21; Ex. 1119 ¶ 145 ("When the source volume consists of a database (as Edwards teaches), a 'data block' is a unit of data used by a database.")). Petitioner asserts, therefore, Edwards teaches a "source database comprising a plurality of database blocks" as recited in claim 1. *See id.*

The next step requires storing database blocks for different point-in-time copies of the source database: "storing on a storage system, database blocks for a plurality of different point-in-time copies of the source database, wherein at least some of the stored database blocks are associated with multiple point-in-time copies of the source database." Petitioner acknowledges Edwards does not disclose explicitly the SnapMirror destination storing multiple snapshots or point-in-time copies received from the source system. *Id.* at 39. To satisfy this limitation, Petitioner asserts, citing the testimony of Dr. Zadok, Patterson discloses a storage system that receives and stores a plurality of snapshots or point-in-time copies of a source volume. *Id.* (citing Ex. 1104, 7, 10; Ex. 1119 ¶ 148). Petitioner also acknowledges Edwards does not disclose explicitly "multiple point-in-time copies [that] share (point to) common data blocks—*i.e.*, that a data block is associated with multiple point-in-time copies." *Id.* at 39–40. Petitioner asserts that Patterson discloses "the SnapMirror destination stores data blocks of multiple snapshots, wherein some of the data blocks are associated with a plurality of snapshots" because Patterson describes "an example where two different point-in-time snapshots of a volume are mirrored to a destination storage system, and those point-in-time copies share at least some of the same stored data blocks." *Id.* at 40 (citing Ex. 1104, 11, Fig. 1). Relying on the Declaration of Dr. Zadok, Petitioner argues this disclosure in Patterson teaches "storing multiple data blocks on a storage system, wherein at least some of the stored data blocks are associated with multiple point-in-time copies." *Id.* at 41 (citing Ex. 1119 ¶ 149). Specifically, similar to our discussion of Patterson above, Petitioner argues that Patterson describes, after an update transfer subsequent to a file system change, blocks B and D stored in the SnapMirror destination would be associated with at least two point-in-time copies, i.e., Base Reference Snapshot and Incremental Reference Snapshot, of the source file system. *Id.* at 22–24 (citing Ex. 1104, 11, Fig. 1). An excerpt of Figure 1 of Patterson provided by the Petitioner is reproduced below.



Id. at 23. The excerpt of Figure 1 above shows that, upon Update Transfer, blocks B and D stored in the SnapMirror destination are associated with two point-in-time copies—the Base Ref Snapshot and Inc Ref Snapshot. *See id.* Quoting Patterson, Petitioner argues "SnapMirror transfers the blocks for all existing snapshots that were created between the base and incremental reference snapshots. . . . Thus, ***the destination has a copy of all of the source's snapshots***." *Id.* at 23–24 (emphasis by Petitioner) (quoting Ex. 1104, 10) (internal quotation marks omitted).

The next step requires creating a set of files for a virtual database. Petitioner asserts Edwards teaches "creating a set of files for a virtual database, each file in the set of files is linked to the database blocks on the storage system," as recited in claim 1, because Edwards describes creating a clone, called a FlexClone volume, from a snapshot, including a snapshot stored on a destination storage. *Id.* at 40–41. A FlexClone volume is essentially a writable version of a FlexVol volume. *See id.* at 21, 41 (citing Ex. 1103, 9, 15). According to Petitioner, Edwards explains that, to create a clone volume, WAFL creates the files required for a new FlexVol volume. *Id.* at 41 (citing Ex. 1103, 15). A FlexVol volume is an "instantiation" of NetApp's file system described in Edwards. *Id.* at 16 (citing Ex. 1103, 10). In general, each file in a FlexVol volume comprises an inode, which contains pointers to the data blocks for the file. *Id.* at 16–17 (citing Ex. 1103, 10). According to Petitioner, these data structures form a tree with a root block called the "vol_info" block, which contains the inodes for all other files in the file system. *Id.* at 17. When creating a clone, a container file for the new clone volume is created and seeded "with a vol_info block that is a copy of the vol_info block of the snapshot copy on which the clone is based." *Id.* at 41 (quoting Ex. 1103, 15). The new vol_info block, like all other vol_info blocks, "***contains the inodes for all of the other files in the file system***, including the other metadata files." *Id* at. 21 (citing Ex. 1103, 10). Hence, Petitioner argues, the cloned volume inherits all the pointers or links of the snapshot copy that point to the underlying data blocks. *Id.* at 41. Relying upon the testimony of Dr. Zadok, Petitioner asserts that Edwards, therefore, describes creating a new vol_info block that points or links to the tree of blocks of the snapshot (i.e., point-in-time) copy, which teaches creating a new, cloned volume (i.e., a new file system (*id.* at 21)) comprising a set of database files linked to or pointing to already-stored database blocks associated with a point-in-time copy of the source database. *Id.* at 41–42 (citing Ex. 1119 ¶ 152).

Citing the testimony of Dr. Zadok, Petitioner asserts "[t]his is the same way that the '808 patent creates files for a virtual database, by implementing them 'as pointers to the actual database block that stores the data.'" *Id.* at 21 (citing Ex. 1101, col. 18, ll. 47–49; Ex. 1119 ¶ 76). According to Petitioner, a FlexClone volume described in Edwards is a "***full-fledged FlexVol volume with all the features and capabilities of a normal WAFL volume***." *Id.* at 21 (emphasis by Petitioner) (quoting Ex. 1103, 15). However, unlike the traditional FlexVol or WAFL volumes, the files in a FlexClone volume are writable files. *Id.* at 21 (citing Ex. 1103, 9).

Relying further upon the testimony of Dr. Zadok, Petitioner asserts that "the virtual database disclosed by Edwards is virtual in that the physical implementation of the database files is decoupled from their logical use" (*id.* at 42 (citing Ex. 1119 ¶ 156)) because a FlexClone volume features "a level of indirection, or ***virtualization***, between the logical storage space used by a volume and the physical storage space provided by the RAID subsystem" (*id.* (emphasis by Petitioner) (quoting Ex. 1103, 11)). Petitioner argues the FlexVol container file includes a "container map" that provides VVBN-to-PVBN mappings that "implement[s] a level of indirection between physical storage containers (called aggregates) and logical volumes (FlexVol volumes)." *Id.* at 18 (quoting Ex. 1103, 10); *see also* Ex. 1103, 12 ("We refer to the VVBN-to-PVBN mapping provided by this first level of indirect data in the container file as the *container map*."). Relying upon the Declaration of Dr. Zadok, Petitioner asserts this virtualization by VVBN-to-PVBN mapping achieves the same decoupling of the physical implementation of the database files from the logical use of the database files described in the '808 patent. Pet. 19 (citing Ex. 1119 ¶ 71; Ex. 1101, col. 5, ll. 18–21).

The next and the last step of claim 1 recites "mounting the set of files associated with the virtual database on a database server allowing the database server to read from and write to the set of files." Petitioner acknowledges Edwards does not explicitly disclose the steps for mounting of a virtual database to a database server. *Id.* at 43. Petitioner asserts Sanders teaches this limitation because Sanders describes that access to a clone database—which comprises a set of files for a virtual database as discussed above—can be provided by mounting the clone volume to a database server. *Id.* at 43–44 (citing Ex. 1105, 14, 28). The record supports, and we adopt, Petitioner's contentions as summarized above.

(2) Patent Owner's Response

Patent Owner in its Response argues that several limitations of claim 1 are not rendered obvious by the proposed combination of Edwards, Patterson, and Sanders. We consider each of those arguments in turn.

(i) Database Blocks

Citing the testimony of Dr. Shenoy, Patent Owner asserts that Edwards and Sanders do not teach the "database blocks" recited in the challenged claims because "database blocks" are used *only* by database servers *at the application level* and the WAFL or FlexVol data blocks described in the prior art are not used by a database server. PO Resp. 40–41 (citing Ex. 2111 ¶¶ 47, 122, 123 ("Database blocks are data structures used by database management systems *at the application level* . . . . A *database server* knows nothing about file system data blocks because it *only* interacts with database blocks and does not directly manipulate file system data blocks."), 125 ("The database blocks [are] used *only by the database server at the application level*."), 126–35) (emphases added). Patent Owner further relies upon the testimony of Dr. Shenoy to assert that the WAFL file system data blocks are not "database blocks" because "[a] file system does not operate *at the application level*" and "does not . . . use database blocks." PO Resp. 41–42 (emphasis added) (citing Ex. 2111 ¶ 124). Dr. Shenoy cites Oracle technical documents (Ex. 2111 ¶¶ 123–124 (citing Ex. 2102), 125 (citing Ex. 2101)) describing the features of "the Oracle database server, an object-*relational* database management system" (Ex. 2101, 25; Ex. 2102, 19) (emphasis added) in support of his testimony. Patent Owner further argues that “database blocks” are used at the application level because the ’808 patent “repeatedly describes” that database blocks are accessed through APIs. PO Resp. 43 (citing Ex. 1101, col. 7, ll. 41–64; col. 10, ll. 15–17).

Patent Owner’s arguments primarily turn on the claim construction of the term “database block” and do not persuasively refute Petitioner’s contentions. As discussed above in Section III.A, we construe the term “database block” to mean “a unit of data used by a database.” As discussed in Section III.B.1 in the context of our claim construction analysis, a database may be used by any application or program, not just a database server application. Hence, Patent Owner’s argument limiting “database blocks” to those that can be used only by database servers is unpersuasive. Similarly, Patent Owner’s “application level” arguments are not persuasive because they essentially rehash Patent Owner’s unpersuasive claim construction arguments that “database blocks” are application-level constructs. As discussed above in Section III.A.2, those arguments by Patent Owner do not comport with the claim language and the written description of the ’808 patent. Furthermore, as discussed in Section III.A.3, the “database blocks” recited in the challenged claims encompass storage level “data blocks.” Patent Owner admits that the terms “file system block” or “data blocks” refer to blocks of data stored on a disk and manipulated by the file system. PO Resp. 24 n.7. Hence, the recited “database blocks” do not exclude file system data blocks, such as the WAFL data blocks. In addition, to the extent Patent Owner argues that the “database blocks” recited in the challenged claims are limited to database blocks used in relational database systems, such as the Oracle systems relied upon by Dr.

Shenoy, Patent Owner's argument is unpersuasive for the reasons discussed above in Sections III.A.1.d and III.B.1.

Next, Patent Owner argues the WAFL file system data blocks are not database blocks because a file system is not a database. *Id.* at 42 (citing Ex. 2111 ¶¶ 131–32). Patent Owner's argument is not persuasive in view of Patent Owner's proposed definition of a "database." In related Case IPR2015-00128, Patent Owner proposed that a plain and ordinary meaning of the term "database" is "a collection of data that is organized so that it can be easily accessed, managed or updated." *Actifio, Inc. v. Delphix, Corp.*, Case IPR2015-00128, slip op. at 33 (PTAB July 28, 2015) (Paper 17) ("'128 PO Resp."). In this case, Patent Owner's expert, Dr. Shenoy, proposes the same definition. Ex. 2111 ¶ 33. However, a file system may also be "a collection of data that is organized so that it can be easily accessed, managed or updated." In fact, Dr. Shenoy acknowledges file systems also "manage data" but in a "less-structured way." *Id.* We find such differences cannot support any patentable distinction in this case. In addition, the '808 patent describes broadly that "[a] database comprises data stored in a computer for use by computer implemented applications." Ex. 1101, col. 4, l. 67–col. 5, l. 1. We do not find any meaningful distinction between a "database" and a "file system" under this description, either. Therefore, we find, in the context of this case, the term "database" encompasses file systems.

Patent Owner also asserts that Petitioner "does not even identify any description of a database using WAFL data blocks." PO Resp. 42. Patent Owner's argument is not persuasive for the reasons discussed above. Furthermore, Petitioner relies on a WAFL or FlexVol volume that stores or consists of a database, such as IBM's DB2 database. *See* Pet. 31, 38 (citing Ex. 1119 ¶ 146). Patent Owner further argues that Edwards does not disclose receiving units of data used by a database. PO Resp. at 40. Patent Owner's argument is unpersuasive because Petitioner relies on the combination of Edwards and Sanders to teach receiving DB2 data blocks at a destination. *See* Pet. 31 ("a [person of ordinary skill in the art] would have readily referred to Sanders to apply the cloning technique described in Edwards to clone DB2 databases"). Nonobviousness cannot be established by attacking the references individually when the unpatentability challenge is based on a combination of prior art disclosures. *See In re Merck & Co. Inc.*, 800 F.2d 1091, 1097 (Fed. Cir. 1986).

Lastly, Patent Owner's arguments based on its proposed claim construction that a "database block" must necessarily include metadata (PO Resp. 43–46) are unpersuasive because we decline to adopt Patent Owner's interpretation for the reasons discussed above in Section III.A.

In view of the foregoing and based on the record before us, we credit the testimony of Dr. Zadok (e.g., Ex. 1119 ¶ 145) and find that data blocks of a WAFL or FlexVol volume that stores or consists of a database, such as IBM's DB2 database, are units of data used by a database. Therefore, the proposed combination of Edwards and Sanders teaches "database blocks."

(ii) Receiving and Storing Different Point-In-Time Copies of a Source Database

Patent Owner argues Edwards does not teach receiving point-in-time copies of a source database comprising database blocks because the

Snapshot copies received at a destination include file system data blocks, not database blocks. PO Resp. 46. Again, Patent Owner's argument turns on claim construction and does not persuasively refute Petitioner's showing because, as discussed above, the disclosure in Edwards or the combination of Edwards and Sanders of data blocks of a WAFL or FlexVol volume that stores or consists of a database teaches "database blocks."

Next, Patent Owner asserts that the proposed combination does not teach receiving "different point-in-time copies of a source database." *Id.* at 46–47. According to Patent Owner, SnapMirror is a mirroring method for disaster recovery, where the source file system and backup "mirror" are always maintained in the identical current state. *Id.* at 47. Patent Owner argues that the SnapMirror "does not create or load a plurality of point-in-time copies of the FlexVol volume it is cloning, it is merely cloning the current state of the data that is within the volume." *Id.*

Patent Owner's argument is unpersuasive because Petitioner relies on SnapMirror's update transfer function described in Patterson to teach receiving and storing a plurality of snapshots or point-in-time copies of a source volume. Pet. 39–40 (citing Ex. 1104, 7, 10; Ex. 1119 ¶ 148), 22–23 (citing Ex. 1104, 10–11, Fig. 1). Nonetheless, Patent Owner argues Patterson does not teach the limitations at issue because the base reference and incremental reference snapshots described in Patterson do not teach receiving point-in-time copies or storing database blocks for the point-in-time copies because "[t]hese temporary snapshots are created at the *source* . . . and are *not* transferred to or stored at the destination." PO Resp. 48. According to Patent Owner, the incremental snapshot of Patterson is a snapshot of the *entire* source system that is "never received by or stored at the *destination*" but, rather, is compared to a base reference snapshot to determine "the correct set of data blocks to transfer to the destination." *Id.*

Patent Owner does not dispute, however, those "correct sets of data blocks" received and stored at the destination teach receiving point-in-time copies of a source database and storing database blocks for the point-in-time copies recited in the challenged claims. As discussed in the Petition, Petitioner relies upon Patterson's incremental update function of transferring to the destination the set of blocks changed since the last update to teach these limitations at issue. Pet. 22–24 (citing Ex. 1104, 10–11, Fig. 1), 39–40; *see also* Pet. Reply 20 ("[A]fter a SnapMirror *update*, 'the destination has a copy of *all of the source's snapshots*.'") (first emphasis added) (citing Ex. 1104, 10). In fact, the '808 patent describes a similar process of copying only the blocks changed since the last update as making point-in-time copies of a production database. *See* Ex. 1101, col. 18, ll. 35–43. Therefore, the record supports Petitioner's showing, and Patent Owner's argument does not persuasively refute Petitioner's contentions. Based on the foregoing, we find the proposed combination teaches "receiving different point-in-time copies of a source database" and "storing . . . database blocks for a plurality of different point-in-time copies of the source database" recited in the challenged claims.

(iii) Virtual Database

Patent Owner asserts that the vol_info block of the FlexClone volume relied upon by Petitioner to teach a virtual database is not a set of database files because the inodes contained in the vol_info block do not point to database blocks, but, rather, point to WAFL file system data blocks. PO Resp. 51, 56. Patent Owner's argument again turns on claim construction and does not persuasively refute Petitioner's showing because, as discussed above, the prior art's disclosure of data blocks of a WAFL, FlexVol, or FlexClone volume that stores or consists of a database teaches "database blocks."

Patent Owner next asserts that neither FlexClone nor FlexVol virtualize databases because they, instead, virtualize disk volumes. *Id.* at 52–53. Patent Owner argues the FlexVol volumes do not virtualize databases because they do not decouple "the physical implementation of the *database files* . . . from the logical use of the *database files*." *Id.* at 53. Patent Owner also argues that the FlexVol volumes operate at the physical storage level, not the application level, and, therefore, have no knowledge of database blocks. *Id.*

Patent Owner's arguments largely turn on the claim construction and do not persuasively refute Petitioner's showing. Patent Owner's arguments regarding virtualization turn in part on the claim construction of the terms "database blocks" and "database." These arguments are unpersuasive for the reasons discussed above in Sections III (Claim Construction) and V.A.5(2)(i) (addressing Patent Owner's arguments regarding "database blocks").

As discussed in Section III.B.2, based on the disclosure in the '808 patent describing the structure of virtual database files, "decoupling of physical implementation of the database files" from "the logical use of database files" can be understood as reading from or writing to database blocks by reading from or writing to virtual database file structures that point to the actual database blocks. Thus, we agree with Petitioner that the "virtualization" described in Edwards using a vol_info block and a container map of VVBN-to-PVBN mappings that point to WAFL data blocks achieves the same decoupling of the physical implementation of the database files from the logical use of the database files described in the '808 patent. *See* Pet. 19 (citing Ex. 1119 ¶ 71; Ex. 1101, col. 5, ll. 18–21).

Patent Owner further argues that our preliminary construction of "virtual database" is so broad as to encompass a "source database." PO Resp. at 50. As discussed above in Section III.B.2, we address Patent Owner's argument by expressly requiring "database blocks" in our construction of the term "virtual database" to be associated with another database. That is, we construe "virtual database" to mean "a set of readable and writable database files capable of being mapped to physical addresses for stored database blocks associated with another database."

The term "database files" used in our claim construction does not need further interpretation because our discussion in Section III.B shows that we used "database file" as "a file used by a database or a database application." For example, we discussed that a "database file" may be used by any application or program. As also discussed in the same section, Patent Owner acknowledged at the Oral Hearing that "a database server" in its proposed construction "is used synonymously . . . with database application." Tr. 184:1–185:12; *see also* Ex. 1101, col. 5, ll. 1–4 ("database server is a computer program that can interact with the database").

Furthermore, as discussed above in Section V.A.5(2)(i), we find, in this case, the term "database" encompasses file systems. Therefore, we find the vol_info block, the inodes, and the container file of a FlexClone or FlexVol volume (which is a file system, as discussed above) relied upon by Petitioner constitute "database files," that is, files used by a database or a database application, at least in the cases where the file system stores a database. Accordingly, we find the new vol_info block of the FlexClone volume that points to the tree of blocks of a point-in-time snapshot copy of a FlexVol volume storing a DB2 source database relied upon by Petitioner teaches a "virtual database," i.e., "a set of readable and writable database files capable of being mapped to physical addresses for stored database blocks associated with another database." The data blocks of the point-in-time snapshot copy of the FlexVol volume storing a DB2 source database are "associated with another database," i.e., the DB2 source database.

(iv) Creating a Set of Files for a Virtual Database

Patent Owner argues creating a "clone vol_info block" is not "creating a set of files" because the new vol_info block is "merely a copy" of the vol_info block of the volume to be cloned. PO Resp. 54–55. Patent Owner's argument is unpersuasive because the challenged claims recite "creating a set of files" and do not require "creating a *new* set of files." In fact, the '808 patent describes creating a virtual database from a copy of another virtual database. Ex. 1101, col. 19, ll. 22–25 ("Point-in-time *copies* of VDB1 are also made based on a predefined schedule. This allows a user to *create* a second virtual database VDB2 based on a point-in-time *copy* of VDB1.") (emphases added).

Next, Patent Owner argues that the "clone vol_info block" is not a set of files because it "merely points to the *existing* files" in the vol_info block of the volume being cloned. PO Resp. 55 (citing Ex. 2111 ¶ 156). Patent Owner explains that according to a dictionary source, an applicable meaning of "**set**" is "[a] group of things of the same kind that belong together and are so used." *See id*. at 55 (citing Ex. 2119, 3–4).

Contrary to Patent Owner's arguments, creating a new "clone vol_info block" creates a new set of files even under Patent Owner's proffered definition of "set" because creating a new member, "clone vol_info block," which points to the tree of blocks of the point-in-time snapshot copy, constitutes creating a new "group of things of the same kind that belong together and are so used." By analogy, the set of numbers (1, 5, 6) is a different set than the set of numbers (2, 1, 5, 6), even though the sets overlap.

In addition, Petitioner argues FlexClone creates a new FlexVol volume: "[A] FlexVol volume is a file system created *within* a file on an underlying file system." Pet. Reply 20 (quoting Ex. 1103, 14) (citing Ex. 1103, 11); Ex. 1133, 257:4–6 ("[A] FlexVol is based on the WAFL file system. So in this case, there is a file system associated with the volume.")). According to Petitioner, the clone vol_info block represents a "file structure" similar to the "'read/write file structure representing a VDB' created in the patent." Pet. Reply 21 (quoting Ex. 1101, col. 18, ll. 27–28). Therefore, Petitioner argues, "the creation of a new vol_info block creates a new file system with new files." *Id*. (citing Ex. 1119 ¶¶ 74–76, Ex. 1166 ¶¶ 59).

As described above in the summary of Petitioner's contentions and the summary of Edwards, the record supports Petitioner. For example, Edwards discloses that

> [c]reating a clone volume is a simple process. WAFL *creates the files required for a new FlexVol volume*. But rather than creating and writing a new file system inside the volume, WAFL seeds the container file of the clone with a vol_info block that is a copy of the vol_info block of the Snapshot copy on which the clone is based.

Ex. 1103, 15 (emphases added). In other words, according to Edwards, WAFL "creates the files" for a new file system during cloning, as Petitioner and Dr. Zadok argue, even if it does not write a new file system inside the volume. *See id*.

Petitioner further argues the files of a FlexClone are writable and, therefore, different than the files of a snapshot's read-only files. Pet. Reply 22 (citing Ex. 1166 ¶¶ 60–61). According to Petitioner, Dr. Shenoy agrees that the files of a FlexClone are writable and the files of a snapshot are read-only. *Id*. (citing Ex. 1133, 268:9–10). Therefore, Petitioner argues creating a FlexClone creates a "new, independent set of files." *Id*. We agree with and adopt Petitioner's contentions.

Based on the foregoing and the record before us, we find creating the new vol_info block of the FlexClone volume that points to the tree of blocks of a point-in-time snapshot copy of a FlexVol volume storing a DB2 source database relied upon by Petitioner teaches "creating a set of files for a virtual database" recited in the challenged claims.

(3) Conclusion

Accordingly, on this record, we find that Petitioner has demonstrated, by a preponderance of evidence, that the proposed combination of Edwards, Patterson, and Sanders teaches every limitation of claim 1.

b. Reasons for Combining Edwards, Patterson, and Sanders

If all elements of a claim are found in the prior art, as is the case here, the factfinder must further consider the factual questions of whether a person of ordinary skill in the art would be motivated to combine those references, and whether in making that combination, a person of ordinary skill would have had a reasonable expectation of success. *Dome Patent L.P. v. Lee*, 799 F.3d 1372, 1380 (Fed. Cir. 2015). Although "there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness," *KSR*, 550 U.S. at 418 (quoting *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006)), "[t]he obviousness analysis cannot be confined by a formalistic conception of the words teaching, suggestion, and motivation, or by overemphasis on the importance of published articles and the explicit content of issued patents." *Id.* at 419. Rather, courts must take an "expansive and flexible approach" to the question of obviousness. *Dome Patent L.P.*, 799 F.3d at 1380 (quoting *KSR*, 550 U.S. at 415).

We find that the Petition provides ample reasons why one of ordinary skill in the art would have combined the teachings of Sanders, Edwards, and

Patterson. Pet. 28–32, 40, 43–44. For example, all of the references address the same problem addressed by the ’808 patent.

According to Patent Owner, “a main purpose of the invention” of the ’808 patent is “to efficiently provide virtual databases at arbitrary points in time without proliferating redundant copies of database data and storage infrastructure.” PO Resp. 17 (citing Ex. 2111 ¶¶ 63–67). The ’808 patent also describes the problems in the existing database solutions as “[m]aking copies of the production databases” for each stage of workflows, which “requires redundant and expensive hardware infrastructure as well as the time overhead required to copy the data” in the BACKGROUND section (Ex. 1101, col. 1, ll. 35–39) and states in the SUMMARY section “[t]o address the needs unmet by existing database technologies, embodiments of the invention enable virtual databases that efficiently use storage and other computing resources.” *Id.* at col. 1, ll. 63–65. The ’808 patent further states as follows:

> Making copies of large databases by conventional means can be a slow process. Furthermore, running different copies of databases on different machines results in inefficient usage of the hardware. Various workflow scenarios associated with databases can be simplified and made highly efficient by creating virtual databases instead of making physical copies of the databases. Multiple virtual databases can be stored in a database storage system 100 and the available resources of the system can be utilized efficiently.

Id. at col. 20, ll. 7–17. Hence, a central problem addressed by the ’808 patent is to avoid making redundant physical copies of data and provide efficient use of storage and other computing resources. The solution described in the ’808 patent is to “creat[e] virtual databases instead of making physical copies of the databases.” *Id.*

Petitioner submits that Edwards, Patterson, and Sanders all solve the same problem—i.e., “the costs and complexities involved in making additional physical copies of databases for experimental, test, developmental, and other purposes”—with the same solution, i.e., virtual databases using FlexClone technology. Pet. 30 (citing Ex. 1119 ¶¶ 108–110). The record supports Petitioner’s contentions. *See* Ex. 1103, 9–11, 14; Ex. 1104, 10; Ex. 1105, 1, 3, 5. Our reviewing court has made clear that “a court . . . may find a motivation to combine prior art references in the nature of the problem to be solved.” *ABT Sys., LLC v. Emerson Elec. Co.*, 797 F.3d 1350, 1360 (Fed. Cir. 2015) (quoting *Ruiz v. A.B. Chance Co.*, 357 F.3d 1270, 1276 (Fed. Cir. 2004)); *see also id.* at 1360–61 (finding motivation to combine because the nature of the problem to be solved is the same in each of the cited references).

Petitioner provides other and similar reasons for combining the various aspects of the NetApp technologies at various places in the Petition. *See* Pet. 27–32, 40, 43–44.

Patent Owner asserts that Edwards, Patterson, and Sanders do not address the problem addressed by the ’808 patent, which is to “run a different version of the database server and/or a different operating system compared to the production database system 110 that is the source of the database being virtualized.” PO Resp. 31 (citing Ex. 1101, col. 6, ll. 59–62). Patent Owner argues one of ordinary skill in the art would not, after reading Edwards, Patterson, and Sanders, know how to put together a system where a virtual database can run on a different database server and operating system than the production database. *Id.* Patent Owner's argument is unpersuasive because the described features are recited only in dependent claims 22 and 23, which are not challenged in this case. Claim 1, from which these claims depend, is presumed to be broader and not limited by the additional limitations recited in these dependent claims relating to running a virtual database on a different database server and operating system than the production database. *Columbia Univ.*, 811 F.3d at 1370. Moreover, a claim is not required to encompass all of the advantages or purposes of the invention. *See Howmedica Osteonics*, 540 F.3d at 1345. Therefore, the features argued by Patent Owner cannot be the basis for rejecting the proposed combination of Edwards, Patterson, and Sanders for purposes of obviousness analysis for the challenged claims.

Patent Owner argues that Edwards, Sanders, and Patterson do not describe the "same" technology, and that "Petitioner's sole argument as to why the documents render a claim limitation obvious is because, Petitioner maintains, they describe 'the same' technology." PO Resp. 4–5. Patent Owner asserts that "FlexVol, FlexClone and SnapMirror are NetApp trademarks . . . associated with products and technologies that changed over time," showing that the documents describe different embodiments. *Id.* at 5.

As Petitioner argues, the references focus on describing the same basic NetApp system, including FlexVol, FlexClone, and SnapMirror, all to provide database cloning, eliminating the need for physical copying, and providing savings in storage space, cost, data recovery, etc. *See* Pet. 28–30. Even if the products did change over time, as Patent Owner contends, understanding the evolution of the underlying related systems reasonably would have been beneficial to better understand the products and apply them for cloning solutions, data back-up, etc. *See, e.g.*, Ex. 1119 ¶ 117 ("Patterson, Edwards, and Sanders all describe the natural evolution of technologies at NetApp") ¶ 114 (starting with Edwards for cloning virtual databases, "the most obvious place to start would be to look at other NetApp publications") ¶¶ 108–111 (Edwards, Sanders, and Patterson address the same problem and solution and provide motivation for combining—e.g., creating multiple virtual databases copies, and providing writable virtual copies, in order to avoid the cost and complexity in making physical copies and to provide data for experimental, test, developmental purposes).

In view of the foregoing, we find the record supports, and Patent Owner's arguments do not persuasively refute, Petitioner's showing, which provides ample motivation and reasons to combine Edwards, Patterson, and Sanders.

c. Conclusion

Upon considering all of the evidence of record, we determine that Petitioner has demonstrated, by a preponderance of evidence, that the subject matter of claim 1 would have been obvious under 35 U.S.C. § 103(a) based on the combination of Edwards, Patterson, and Sanders.

6. Claim 50

Claim 50 recites a computer processor and a computer-readable storage medium storing computer program modules including: (a) a point-in-time copy manager module configured to: receive different point-in-time copies of a source database, the source database comprising a plurality of database blocks; (b) a storage allocation manager module configured to: store on a storage system, database blocks for a plurality of different point-in-time copies of the source database, wherein at least some of the stored database blocks are associated with multiple point-in-time copies of the source database; (c) a virtual database manager module configured to: create a set of files for a virtual database, each file in the set of files linked to the database blocks on the storage system associated with a point-in-time copy of the source database; and (d) a file sharing manager module configured to: mount the set of files associated with the virtual database on a database server allowing the database server to read from and write to the set of files. Petitioner notes that the steps to be performed by the four claimed modules recited by claim 50 (i.e., "point-in-time copy manager module," "storage allocation manager module," "virtual database manager module," and "file sharing manager module") are essentially the same as the steps required by claim 1. Pet. 45–46. Thus, Petitioner asserts that the combination of Edwards, Sanders, and Patterson renders these four "module limitations" of claim 50 obvious for the same reasons that the same combination of prior art references renders obvious the corresponding limitations of claim 1. *Id.* at 46 (citing Ex. 1119 ¶ 176).

Petitioner further asserts that the computer program modules recited in the claim represent software code embodied on a machine-readable medium. *Id.* (citing Ex. 1101, col. 31, ll. 50–53; Ex. 1119 ¶ 178). Relying upon the testimony of Dr. Zadok, Petitioner argues the snapshot, SnapMirror, FlexVol, and FlexClone features of the NetApp technology as described in Edwards, Patterson, and Sanders are all features implemented in code that was programmed to perform these NetApp features. Pet. 46–47 (citing Ex. 1119 ¶¶ 178–179). Thus, Petitioner asserts the computer program module limitations of claim 50 are obvious over the combination of Edwards, Sanders, and Patterson for the same reasons that the same combination of prior art references renders obvious the corresponding limitations of claim 1. *Id.*

Patent Owner argues claims 1 and 50 together and does not respond specifically to Petitioner's showing regarding claim 50 beyond Patent Owner's arguments advanced with respect to claim 1 discussed above. *See* PO Resp. 39–56.

In our Scheduling Order, we cautioned Patent Owner that any arguments for patentability not raised in the response will be deemed waived. *See* Paper 12, 3. Patent Owner has elected not to respond separately, with specificity, to the ground of unpatentability asserted against claim 50. Upon considering all of the evidence of record, for the reasons explained above with respect to claim 1, we determine that Petitioner has demonstrated, by a preponderance of evidence, that the subject matter of claim 50 would have been obvious under 35 U.S.C. § 103(a) based on the combination of Edwards, Patterson, and Sanders.

7. Dependent Claims 3, 29, 31, 53, 54, and 56 (IPR2015-00016)

For our analysis of claims 3, 29, 31, 53, 54, and 56, which depend from claims 1 or 50, we apply our findings and conclusions regarding claims 1 and 50 discussed above. For example, as discussed above in Section

V.A.5.a, we find Petitioner has established, by a preponderance of evidence, that the combination of Edwards, Sanders, and Patterson teaches "receiv[ing] different point-in-time copies of a source database" as recited in claims 1 and 50. Furthermore, we find one of ordinary skill in the art would have been motivated, or would have had a sufficient reason, to combine Edwards, Sanders, and Patterson to teach all the limitations of base claims 1 and 50.

a. Petitioner's Contentions

Claims 3, 29, and 31 depend from claim 1 and further recite "sending a request to receive a point-in-time copy of a source database," "receiving point-in-time copies comprises receiving data streams corresponding to the point-in-time copies, wherein each data stream comprises data from database blocks associated with the source database," and "a data stream corresponding to a first point-in-time copy includes database blocks changed in the source database since a previous point-in-time copy was received," respectively. Relying upon the testimony of Dr. Zadok, Petitioner has shown Patterson teaches the additionally recited limitations of claims 3, 29, and 31. Pet. 48–51 (citing Ex. 1104, 9, 10, 12; Ex. 1119 ¶¶ 162, 163, 165, 166). Petitioner provides detailed explanations supported by the testimony of Dr. Zadok and specific citations to Patterson indicating where in Patterson the claimed features are taught. Relying further upon the Declaration of Dr. Zadok, Petitioner asserts that a person of ordinary skill would have been motivated to combine Edwards with Patterson to teach these additional limitations and provides specific rationales for the combination to teach each of the additional limitations. *Id.* at 49, 50, 51 (citing Ex. 1119 ¶¶ 164, 166, 170).

For example, regarding claim 3, Petitioner asserts that Patterson teaches "sending a request to receive a point-in-time copy of a source database" because Patterson's destination system "triggers SnapMirror updates" and "initiates the mirror relationship ***by requesting*** an initial transfer from the source." Pet. 48 (citing Ex. 1104, 10). According to Petitioner, the source "responds by taking a base ***reference snapshot and then transferring*** all the blocks that are allocated in that or any earlier snapshot, as specified in the snapshots' active map files." *Id.* (citing Ex. 1104, 10). Citing the testimony of Dr. Zadok, Petitioner further asserts that Patterson expressly discloses that the source volume can include a database. *Id.* at 49 (citing Ex. 1104, 12; Ex. 1119 ¶ 163).

Relying upon the testimony of Dr. Zadok, Petitioner further asserts that a person of ordinary skill in the art would have been motivated to combine Edwards with Patterson because Edwards expressly refers to Patterson for details of how NetApp's SnapMirror technology works and because Edwards expressly states that a snapshot on a SnapMirror destination can be used to create a clone database. *Id.* (citing Ex. 1119 ¶ 164).

Regarding claim 29, which depends directly from claim 1, Petitioner asserts that Patterson teaches receiving data streams of the point-in-time copies of the source database because Patterson describes performing the initial transfer or the subsequent incremental updates over a TCP/IP connection. *Id.* at 49–50 (citing Ex. 1104, 10, 12). Relying upon the testimony of Dr. Zadok, Petitioner asserts that a person of ordinary skill in the art would have understood that data transmitted through the TCP/IP connection is streamed. *Id.* at 50 (citing Ex. 1119 ¶ 166).

Claim 31 depends directly from claim 29 and further recites "a data stream corresponding to a first point-in-time copy includes database blocks changed in the source database since a previous point-in-time copy was received." Petitioner asserts that Patterson teaches the additionally recited limitation of claim 31 because Patterson describes incremental updates where only "new or modified" blocks are transferred to the destination. *Id.* at 51 (quoting Ex. 1104, 9).

Claims 53, 54, and 56 depend from claim 50 and further recite "the point-in-time copy manager module is further configured to: send a request to receive a point-in-time copy of a source database," "the point-in-time copy manager module receives point-in-time copies comprising data streams, wherein a data stream comprises data from database blocks of the source database," and "the data stream comprises database blocks changed in the source database since a previous point-in-time copy was received," respectively. Petitioner asserts these limitations are essentially the same as the limitations recited in claims 3, 29, and 31, respectively. Pet. 52–54. Thus, relying upon the testimony of Dr. Zadok, Petitioner argues claims 53, 54, and 56 are rendered obvious by the combination of Edwards, Patterson, and Sanders for the same reasons that the same combination of prior art references renders claims 3, 29, and 31 obvious. *Id.* (citing Ex. 1119 ¶¶ 183, 184, 186, 187, 189).

The record supports, and we adopt, Petitioner's contentions as outlined above.

b. Patent Owner's Response

Patent Owner asserts that Patterson's disclosure of requesting an initial transfer relied upon by Petitioner does not teach the recited limitations of claims 3 and 53 because Patterson's initial transfer sends "the source volume *in its entirety*, and not any particular snapshot of the volume." PO Resp. 56–57 (citing Ex. 2111 ¶ 159). According to Patent Owner, there is no teaching in Patterson that the destination system can "selectively" request a "particular" point-in-time copy or snapshot of the source volume. *Id.* at 57 (citing Ex. 2111 ¶¶ 159, 161).

c. Analysis

Patent Owner's argument is unpersuasive because claims 3 and 53 recite sending "a request to receive a point-in-time copy of a source database," not sending "a *selective* request to receive a *particular* point-in-time copy." As Petitioner argues persuasively, the plain meaning of the words "a" or "an" is "one or more." Pet. Reply 24–25 (citing *01 Communique Lab., Inc. v. LogMeIn, Inc.*, 687 F. 3d 1292, 1297 (Fed. Cir. 2012)). Petitioner also argues persuasively that Patent Owner has not presented any evidence why we should depart from the plain meaning and interpret the claims to require "a *selective* request" and "a *particular* point-in-time copy." *Id.* Hence, we interpret the indefinite article "a" recited in claims 3 and 53 to accord it its plain meaning, i.e., "one or more."

Under the plain meaning interpretation, requesting an initial transfer described in Patterson is "sending ***a*** request" and the entire copy of the source transmitted in response is "***a*** point-in-time copy" of the source. In fact, as discussed in both Petitioner's and Patent Owner's summaries of the '808 patent, the '808 patent describes a similar point-in-time update process where a transfer of an entire copy of the source database is initiated based on a first request for data transfer. *See* Pet. 5 (citing Ex. 1101, col. 6, ll. 25–34; col. 12, ll. 42–49); PO Resp. 8–9 (citing Ex. 1101, col. 9, ll. 52–58).

Furthermore, Patent Owner admits that Patterson teaches "sending a request" for subsequent incremental updates. For example, Patent Owner describes the incremental update function of Patterson as follows: "*Updates to the destination* happen on a schedule; when the destination mirror has an update scheduled, it will *inform the source*, which takes a new 'incremental reference snapshot' of the source system at the current time." PO Resp. 37 (citing Ex. 1104, 10) (emphases added). Indeed, the cited portion of Patterson states "[w]hen a mirror has an update scheduled, it *sends a message* to the source." Ex. 1104, 10 (emphasis added).

Patent Owner does not dispute motivation to combine Edwards, Sanders, and Patterson as it relates specifically to the limitations recited in claims 3 and 53 beyond what was discussed in the context of claims 1 and 50. Based on the foregoing, we find the proposed combination teaches "send[ing] a request to receive a point-in-time copy of a source database," as recited in claims 3 and 53.

Patent Owner does not respond separately, with specificity, to Petitioner's challenge to dependent claims 29, 31, 54, and 56 beyond Patent

Owner's arguments advanced with respect to claims 1 and 50 discussed above. As discussed above, in our Scheduling Order, we cautioned Patent Owner that any arguments for patentability not raised in the response will be deemed waived. *See* Paper 12, 3. Patent Owner has elected not to respond separately, with specificity, to the ground of unpatentability asserted against claims 29, 31, 54, and 56.

Accordingly, upon considering all of the evidence of record, we determine that Petitioner has demonstrated, by a preponderance of evidence, that the subject matter of claims 3, 29, 31, 53, 54, and 56 would have been obvious under 35 U.S.C. § 103(a) based on the combination of Edwards, Patterson, and Sanders.

8. Dependent Claims 2, 24–26, and 51 (IPR2015-00019)

Petitioner has shown that the combination of Edwards, Patterson, and Sanders teaches every limitation of claims 2, 24–26, and 51. '019 Pet. 41–47, 51–52. Petitioner points to specific disclosures in the prior art that are deemed to describe or teach all claim limitations. *Id.* In addition, Petitioner relies upon the Declaration of Dr. Zadok (Ex. 1119) to support its positions. *Id.*

Claim 2 depends from claim 1 and further recites

> receiving a read request for data in the set of files associated with the virtual database from the database server;
>
> accessing data in at least one database block associated with a file in the set of files; and
>
> sending the data in response to the read request.

Petitioner asserts that the combination of Edwards and Sanders teaches the recited limitations of claim 2. *Id.* at 41–43 (citing Ex. 1203, 11, 12, 14;

Ex. 1205, 50, 51). Petitioner also relies upon the Declaration of Dr. Zadok to support its contentions. *Id.* at 42–43 (citing Ex. 1219, ¶ 160). Relying further upon the testimony of Dr. Zadok, Petitioner contends a person of ordinary skill would have combined the teachings of Edwards with Sanders's disclosure of accessing a clone database because "both publications describe the same FlexVol and FlexClone technology and describe creation of clone databases for reading and writing clone databases." *Id.* at 42 (citing Ex. 1219, ¶ 159).

Claim 24 depends directly from claim 1, and claims 25 and 26 depend directly from claim 24. Relying upon the testimony of Dr. Zadok, Petitioner asserts that the combination of Edwards, Patterson, and Sanders teaches the additionally recited limitations of claims 24–26. *Id.* at 43–47 (citing Ex. 1203, 11, 14, 15; Ex. 1205, 28; Ex. 1219 ¶¶ 164, 165, 169, 171, 172). Petitioner provides detailed explanations supported by the testimony of Dr. Zadok and specific citations to Edwards with Sanders indicating where in the applied references the claimed features are taught. Relying further upon the Declaration of Dr. Zadok, Petitioner also provides specific rationales for combining Edwards with Sanders to teach the additionally recited limitations. *Id.* at 44, 45 (citing Ex. 1219 ¶¶ 165, 167).

Claim 51 depends from claim 50 and further recites

> wherein the virtual database manager module is further configured to:
>
> receive a read request for data in the set of files associated with the virtual database from the database server;
>
> access data in at least one database block associated with a file in the set of files; and
>
> send the data in response to the read request.

Petitioner contends the three steps recited in claim 51 are essentially the same as the three steps recited in claim 2. Thus, Petitioner contends that the combination of Edwards, Sanders, and Patterson renders claim 51 obvious for the same reasons that the same combination of prior art references renders claim 2 obvious. *Id.* at 51–52.

The record supports, and we adopt, Petitioner's contentions as outlined above.

As noted above, the Scheduling Order cautions Patent Owner that any arguments for patentability not raised in the Response will be deemed waived. *See* Paper 12, 3. Patent Owner has elected not to respond separately, with specificity, to the grounds of unpatentability asserted against claims 2, 24–26, and 51 based on the combination of Edwards, Sanders, and Patterson.

Accordingly, upon considering all of the evidence of record, we determine that Petitioner has demonstrated, by a preponderance of evidence, that the subject matter of claims 2, 24–26, and 51 would have been obvious under 35 U.S.C. § 103(a) based on the combination of Edwards, Patterson, and Sanders.

B. Obviousness Based on the Combination of Edwards, Patterson, Sanders, and Singh (IPR2015-00016)

Petitioner asserts claim 36 is unpatentable under 35 U.S.C. § 103(a) over the combination of Edwards, Patterson, Sanders, and Singh. Pet. 54–55.

1. Singh

Singh describes the same underlying NetApp technologies disclosed in Edwards, Patterson, and Sanders, including NetApp's WAFL file system, SnapMirror, and snapshots. For example, Singh describes replicating data from a source storage system to a destination storage system using SnapMirror. Ex. 1106, col. 7, ll. 2–11.

Singh also describes compressing data from the source storage system and transmitting the compressed data to the destination over a communication link. *Id.* at col. 7, l. 28–col. 8, l. 11. When received at the destination system, the transmitted data is decompressed and stored in a storage media of the destination storage system. *Id.* at col. 8, ll. 15–21.

2. Analysis

Claim 36 depends from claim 1 and further recites "compressing the database blocks prior to storing on the storage system." Petitioner acknowledges there is no express disclosure of compressing data blocks in Edwards, Sanders, or Patterson. Pet. 25. To satisfy this limitation, Petitioner asserts that Singh teaches compressing data blocks prior to transmitting to a SnapMirror destination and storing on the SnapMirror destination. *Id.* at 54; *see id.* at 25–27 (citing Ex. 1106, col. 2, ll. 64–67; col. 6, ll. 55–59; col. 7, ll. 5–10, 19–22). Relying upon the testimony of Dr. Zadok, Petitioner argues, because Singh teaches the data is compressed prior to transmission to a destination storage system, Singh teaches the data is compressed prior to receiving and storing at destination storage system. *Id.* at 27 (citing Ex. 1119 ¶ 99).

Relying further upon the testimony of Dr. Zadok, Petitioner argues it would have been obvious to a person of ordinary skill in the art to combine compression to gain further network bandwidth reduction and storage efficiency, as efficient data transfer and storage efficiency were one of the main objectives in Edwards, Sanders, and Patterson. *Id.* at 54 (citing Ex. 1119 ¶ 173). The record supports, and we adopt, Petitioner's contentions as summarized above.

Patent Owner asserts that Singh does not teach the recited limitation of claim 36 because Singh describes decompressing the data after it is transported to the target system and before it is stored. PO Resp. 57 (citing Ex. 1106, col. 8, ll. 15–21; Ex. 2111 ¶ 162). Patent Owner argues Petitioner's interpretation that claim 36 encompasses compressing then decompressing before storage is "nonsensical" because the '808 patent requires that "the data be stored in *compressed form*." *Id.* at 57–58 (citing Ex. 1101, col. 23, ll. 35–40).

Petitioner responds persuasively that its interpretation is supported by the written description because the '808 patent expressly contemplates the reasons for compressing data for efficient data transmission. Pet. Reply 25 (citing Ex. 1101, col. 25, ll. 60–61). We also agree with Petitioner that claim 36 does not recite "storing compressed database blocks," but, rather, recites "compressing the database blocks prior to storing on the storage system." *Id.* Hence, the record supports, and Patent Owner's argument does not persuasively refute, Petitioner's showing.

Upon considering all of the evidence of record, we determine that Petitioner has demonstrated, by a preponderance of evidence, that the subject matter of claim 36 would have been obvious under 35 U.S.C. § 103(a) based on the combination of Edwards, Patterson, Sanders, and Singh.

C. Obviousness Based on the Combination of Edwards, Patterson, Sanders, and Fair (IPR2015-00019)

As discussed above in Section V.A.8, we found Petitioner has established, by a preponderance of evidence, that the combination of Edwards, Patterson, and Sanders renders the subject matter of base claims 1 and 50 obvious. We also found that the subject matter of claims 24 and 26, from which claim 27 depends, are obvious over the same combination of prior art. Petitioner adds the teachings of Fair to the basic combination of Edwards, Patterson, and Sanders in an asserted ground of obviousness as to dependent claims 2, 27, 35, 51, and 52. ’019 Pet. 52–60.

1. Fair

Fair describes a system and method for creating a writable clone of a read-only volume. Ex. 1206, Abstract. Although Fair describes the same NetApp technologies disclosed in Edwards, Patterson, and Sanders, including NetApp’s WAFL file system, snapshots, and Data ONTAP operating system, Fair provides a more detailed description of how NetApp writes data by creating a clone of a snapshot. *See id.* at col. 2, l. 8–col. 8, l. 21.

2. Dependent Claims 27 and 52

Claim 27 depends directly from claim 26 and further recites as follows, with the disputed limitations emphasized in *italics*:

> wherein at least some of the stored database blocks associated with the first set of files are also associated with the second set of files;
>
> *responsive to the database block being also associated with the second set of files associated with the second virtual database*, copying the database block; and
>
> linking the copied database block with the file and writing the data to the copied database block.

Claim 52 depends directly from claim 50 and further recites as follows, with the disputed limitation emphasized in *italics*:

> wherein the virtual database manager module is further configured to:
>
> receive a request to write data to the first virtual database from the database server;
>
> identify a database block associated with a file in the first set of files associated with the first virtual database;
>
> *responsive to the database block being also associated with the second set of files associated with the second virtual database*, copy the database block; and
>
> link the copied database block with the file and write the data to the copied database block.

Petitioner asserts that claim 27 recites the normal procedure of writing to a data block in a copy-on-write ("COW") file system. '019 Pet. 54. According to Petitioner, a COW system, when writing to a data block, creates a copy of the block and writes to the copied data block instead of overwriting the current version of the data block. *Id.* (citing Ex. 1203, 10; Ex. 1219 ¶ 175). Petitioner asserts Edwards and Fair each teach creating multiple FlexClone volumes, i.e., copies, from the same mirror snapshot of a source volume. *Id.* at 43–44 (citing Ex. 1203, 15; Ex. 1219 ¶ 164), 54 (citing Ex. 1206, col. 21, ll. 26–28; Ex. 1219 ¶ 176). Hence, Petitioner argues that Edwards and Fair teach that "at least some of the stored database blocks associated with the first set of files are also associated with the second set of files," as recited in claim 27. *Id.* at 44, 54. Petitioner further asserts that Fair teaches the "linking" limitation of claim 27, i.e., "linking the copied database block with the file and writing the data to the copied database block." *Id.* at 54–55 (citing Ex. 1206, col. 2, ll. 10–14; col. 5, ll. 19–21).

Patent Owner asserts that Petitioner fails to show how the proposed combination teaches the COW operation is performed "responsive to the database block being also associated with the second set of files associated with the second virtual database," as recited in claims 27 and 52. '019 PO Resp. 56–57 (citing Ex. 2212 ¶ 167). Citing certain cross-examination testimony of Dr. Zadok, Patent Owner asserts that Petitioner concedes that NetApp's copy-on-write technique is applied regardless of whether a block is shared or not. *Id.* at 56 (citing Ex. 2211, 183:24–184:16). In other words, Fair's COW operation is "always on," i.e., it always creates a new copy when a data block is modified.

Petitioner responds with two arguments. First, Petitioner argues, although "claim [27] appears to require" the COW operations "are ***caused by*** the fact that the data block being written to is shared by two virtual databases," Patent Owner's infringement contentions in the related district court cases do not require such a causal relationship. '019 Pet. 55–56 (citing Ex. 1212, 74–75). Petitioner does not explain, however, why the disputed limitation should be interpreted to not require such a causal relationship. Second, Petitioner argues that the '808 patent discloses an embodiment where COW write requests are used on any block regardless of whether that block is shared by two virtual databases or not. ’019 Pet. Reply 25 (’019 Paper 32) (citing Ex. 1201, col. 10, ll. 23–31; Fig. 12). However, the claims expressly recite “*responsive to* the database block being *also* associated with the second set of files associated with the second virtual database.” As discussed above, Petitioner concedes this point. *See* ’019 Pet. 55–56. Petitioner also acknowledges that the ’808 patent describes an embodiment where the COW operations are caused by or performed in response to a block being shared by two virtual databases. *Id.* at 56 (citing Ex. 1201, col. 2, ll. 60–64). Petitioner describes the cited passage of the ’808 patent as “explaining that these steps are performed ‘***[i]f*** the database block identified is *also* associated with the second VDB.’” *Id.* (second emphasis added). Hence, the record supports Patent Owner’s argument and does not support Petitioner’s arguments.

Based on the foregoing, we determine that Petitioner has not demonstrated, by a preponderance of evidence, that the subject matter of claims 27 and 52 would have been obvious under 35 U.S.C. § 103(a) based on the combination of Edwards, Patterson, Sanders, and Fair.

3. Dependent Claims 2, 35, and 51

Claims 2 and 35 depend from claim 1, and claim 51 depends from claim 50. Petitioner has shown Fair teaches the additionally recited limitations of claims 2, 35, and 51. ’019 Pet. 52–53 (claim 2), 56–57 (claim 35), 57–58 (claim 51). Petitioner provides detailed explanations supported by the testimony of Dr. Zadok and specific citations to Fair indicating where in Fair the claimed features are taught. *See id.* Relying further upon the Declaration of Dr. Zadok, Petitioner also provides specific rationales for combining Edwards and Sanders with Fair to teach the additionally recited limitations. ’019 Pet. at 53, 57 (citing Ex. 1219 ¶¶ 161, 181–182). The record supports, and we adopt, Petitioner’s contentions as outlined above.

As noted above, the Scheduling Order cautioned Patent Owner that any arguments for patentability not raised in the Response will be deemed waived. *See* Paper 12, 3. Patent Owner has elected not to respond separately, with specificity, to the grounds of unpatentability asserted against claims 2, 35, and 51 based on the combination of Edwards, Patterson, Sanders, and Fair.

Accordingly, upon considering all of the evidence of record, we determine that Petitioner has demonstrated, by a preponderance of evidence, that the subject matter of claims 2, 35, and 51 would have been obvious under 35 U.S.C. § 103(a) based on the combination of Edwards, Patterson, Sanders, and Fair.

VI. PATENT OWNER’S MOTIONS TO EXCLUDE EVIDENCE

In *inter partes* reviews, documents are admitted into evidence subject to an opposing party asserting objections to the evidence and moving to exclude the evidence. 37 C.F.R. § 42.64. As the moving party, Patent Owner has the burden of showing that an Exhibit is not admissible. 37 C.F.R. § 42.20(c).

A. The ’016 IPR

Patent Owner moves to exclude Petitioner’s Exhibits 1134–1139, 1140–1148, 1150–1153, 1156, 1158–1161, and 1165. PO Mot. to Exclude (Paper 46), 1. Patent Owner contends that Petitioner does not rely on Exhibits 1134–1138, 1140, 1141, 1143, 1147, and 1160. *Id.* at 1 n.1. Except for Exhibits 1138 and 1160, Petitioner does not dispute this contention. *See* Pet. Exclude Opp. (Paper 53). We, therefore, grant Patent Owner’s motion as it pertains to Exhibits 1134–1137, 1140, 1141, 1143, and 1147. Further, Petitioner has moved, unopposed, to expunge Exhibit 1130 (*see* Paper 43), which motion we hereby grant.

Of the objected-to Exhibits, except for Exhibits 1142, 1150, 1151, and 1152, we do not, and need not, consider such evidence. We have determined, as discussed above, that Petitioner has demonstrated by a preponderance of the evidence that the challenged claims are unpatentable, without need for Petitioner’s additional arguments or evidence in relation to those additional Exhibits. Accordingly, Patent Owner’s motion to exclude Exhibits 1138, 1139, 1144–1146, 1148, 1153, 1156, 1158–1161, and 1165 is *dismissed* as moot. We consider Patent Owner’s motion to exclude with respect to each of the remaining exhibits below.

1. Exhibits 1142 and 1151

Exhibit 1142 is U.S. Patent No. 8,468,174 B1 (“the ’174 patent”), which is the subject of related proceeding IPR2015-00128. Exhibit 1151 is a technical document describing Google’s BigTable database. Patent Owner’s Motion does not specify the basis for any evidentiary objection to these

Exhibits. Therefore, Patent Owner's motion to exclude Exhibits 1142 and 1151 is *denied*.

2. Exhibit 1150

Exhibit 1150 is the Supplemental Declaration of Louis Hernandez. Patent Owner argues that the Exhibit is "inadmissible hearsay." PO Mot. to Exclude 8–11. The Supplemental Declaration, however, consists of statements made by Mr. Hernandez while testifying in this proceeding—not "hearsay" (Fed. R. Evid. 801(c)), but sworn testimony that is subject to cross-examination. Indeed, Patent Owner cross-examined Mr. Hernandez with respect to that testimony.[23]

Patent Owner also argues that Mr. Hernandez lacks personal knowledge to testify. Mr. Hernandez's personal knowledge of practices about NetApp document publications comes from his review and recognition of documents published before and during his tenure as an employee of NetApp. *See* Ex. 1122 ¶¶ 1–6; Ex. 1150 ¶¶ 3–11. On this record, ample basis exists for him to testify with personal knowledge of the facts under Rule 602. We are not persuaded otherwise by Patent Owner's

[23] Patent Owner's allegation of "double hearsay" is not persuasive. PO Mot. to Exclude 8–9; PO Exclude Reply 5. Patent Owner does not argue that it objected to any NetApp documents that Mr. Hernandez relied upon as hearsay. *See* Pet. Exclude Opp. 13–14; PO Exclude Reply 5 (replying to Petitioner's contention but not disputing a lack of an objection). In addition, as discussed herein and further below, Mr. Hernandez relies on document dates, other indicia, and his knowledge of NetApp's standard practices about dated NetApp documents, not merely dates on documents. *See* Pet. Reply 5 (citing Ex. 1122 ¶ 6; Ex. 1130 ¶¶ 1–5); Pet. Exclude Opp. 7–14.

characterization of Mr. Hernandez's direct and cross-examination testimony. *See* PO Mot. to Exclude 9–11.

Patent Owner's motion to exclude Exhibit 1150 is *denied*.

3. Exhibit 1152

The only reason Patent Owner gives for excluding Exhibit 1152 is that it is "untimely under 37 C.F.R. § 42.23(b) and 37 C.F.R. § 42.123." PO Mot. to Exclude, 1 n.1. Contrary to this argument, Petitioner cites this document as rebuttal evidence in response to Patent Owner's claim construction (i.e., to show that a flat file database would have no metadata in its blocks). *See* Pet. Reply 8–9 (arguing that Patent Owner's preliminary database block construction before a district court did not include metadata, and that the alleged metadata requirement only occurred after Petitioner filed "the last of its IPR petitions," citing Ex. 1124, 6 (database block construction after e-mail chain of Oct. 23, 2014)); PO Resp. 21 n.4 (citing Ex. 2118, 4 (joint district court claim construction, Oct. 27, 2014), 22 n.5).

Patent Owner's motion to exclude Exhibit 1152 is *denied*.

B. The '019 IPR

Patent Owner moves to exclude Petitioner's Exhibits 1234–1241, 1243–1248, 1250, 1256, 1260, 1264, and 1265.[24] '019 PO Mot. to Exclude

[24] We need not identify the Exhibits in the '019 IPR with the "'019" prefix because all exhibits have been assigned unique numbers across the related IPRs involving the '808 patent.

(Paper 46), 1.[25] Patent Owner contends that Petitioner does not rely on Exhibits 1234–1238, 1240, 1241, 1243, 1247, and 1264. *Id.* at 1 n.1. Except for Exhibit 1238, Petitioner does not dispute this contention. *See* ’019 Pet. Exclude Opp. (Paper 53). We, therefore, grant Patent Owner’s motion as it pertains to Exhibits 1234–1237, 1240, 1241, 1243, 1247, and 1264. Further, Petitioner has moved, unopposed, to expunge Exhibit 1230 (*see* Paper 43), which motion we hereby grant.

Of the objected-to Exhibits, except for Exhibit 1250, we do not, and need not, consider such evidence. We have determined, as discussed above, that Petitioner has demonstrated by a preponderance of the evidence that the challenged claims are unpatentable, without need for Petitioner’s additional arguments or evidence in relation to those additional Exhibits. Accordingly, Patent Owner’s motion to exclude Exhibits 1238, 1239, 1244–1246, 1248, 1256, 1260, and 1265 is *dismissed* as moot. We consider Patent Owner’s motion to exclude Exhibit 1250 below.

[25] Patent Owner also states in a footnote “Exhibits 1242, 1251, 1253, 1256, [and] 1259 should be excluded because they are untimely.” *Id.* at 1 n.1. However, these Exhibits are not identified in Patent Owner’s statement of the relief requested, which states “Patent Owner . . . moves to exclude exhibits 1234-41, 1243-48, 1250, 1256, 1260, 1264, and 12651.” *Id.* at 1. To the extent Patent Owner has moved to exclude Exhibits 1242, 1251, 1253, 1256, and 1259 by mentioning them only in a footnote, Patent Owner’s Motion does not comply with 37 C.F.R. § 42.22(a)(1), which requires “[a] statement of the precise relief requested.” Furthermore, it is not clear if Petitioner has been afforded sufficient notice regarding these Exhibits mentioned in a footnote only because Petitioner’s Opposition does not address them. Therefore, we decline to consider Patent Owner’s argument regarding Exhibits 1242, 1251, 1253, 1256, and 1259.

1. Exhibit 1250

Exhibit 1250 is the Supplemental Declaration of Louis Hernandez. Patent Owner argues that the Exhibit is "inadmissible hearsay." '019 PO Mot. to Exclude 8–11. The Supplemental Declaration, however, consists of statements made by Mr. Hernandez while testifying in this proceeding—not "hearsay" (Fed. R. Evid. 801(c)), but sworn testimony that is subject to cross-examination. Indeed, Patent Owner cross-examined Mr. Hernandez with respect to that testimony.[26]

Patent Owner also argues that Mr. Hernandez lacks personal knowledge to testify. Mr. Hernandez's personal knowledge of practices about NetApp document publications comes from his review and recognition of documents published before and during his tenure as an employee of NetApp. *See* Ex. 1222 ¶¶ 1–6; Ex. 1250 ¶¶ 3–11. On this record, ample basis exists for him to testify with personal knowledge of the facts under Rule 602. We are not persuaded otherwise by Patent Owner's characterization of Mr. Hernandez's direct and cross-examination testimony. *See* '019 PO Mot. to Exclude 9–11.

[26] Patent Owner's allegation of "double hearsay" is not persuasive. '019 PO Mot. to Exclude 8–9; '019 PO Exclude Reply 5. Patent Owner does not argue that it objected to any NetApp documents that Mr. Hernandez relied upon as hearsay. *See* Pet. '019 Exclude Opp. 13–14; '019 PO Exclude Reply 5 (replying to Petitioner's contention but not disputing a lack of an objection). In addition, as discussed herein and further below, Mr. Hernandez relies on document dates, other indicia, and his knowledge of NetApp's standard practices about dated NetApp documents, not merely dates on documents. *See* '019 Pet. Reply 5 (citing Ex. 1222 ¶ 6; Ex. 1230 ¶¶ 1–5); '019 Pet. Exclude Opp. 7–14.

Patent Owner’s motion to exclude Exhibit 1250 is *denied*.

VII. CONCLUSION

Petitioner has met its burden of proof, by a preponderance of the evidence, in showing that claims 2, 3, 24–26, 29, 31, 35, 36, 51, 53, 54, and 56 of the ’808 patent are unpatentable based on the following grounds of unpatentability:

A. Claims 2, 3, 24–26, 29, 31, 51, 53, 54, and 56 under 35 U.S.C. § 103(a) as obvious over the combination of Edwards, Patterson, and Sanders;

B. Claim 36 under 35 U.S.C. § 103(a) as obvious over the combination of Edwards, Patterson, Sanders, and Singh; and

C. Claims 2, 35, and 51 under 35 U.S.C. § 103(a) as obvious over the combination of Edwards, Patterson, Sanders, and Fair.

Petitioner has not demonstrated, by a preponderance of evidence, that the subject matter of claims 27 and 52 would have been obvious under 35 U.S.C. § 103(a) based on the combination of Edwards, Patterson, Sanders, and Fair.

VIII. ORDER

In consideration of the foregoing, it is hereby

ORDERED that proceedings in IPR2015-00016 and IPR2015-00019 are consolidated for purposes of issuance of this final decision;

FURTHER ORDERED that claims 2, 3, 24–26, 29, 31, 35, 36, 51, 53, 54, and 56 of the ’808 patent are *unpatentable*;

FURTHER ORDERED that claims 27 and 52 have *not* been shown to be unpatentable;

FURTHER ORDERED that Petitioner's motions to expunge Exhibits 1130 and 1230 are *granted*;

FURTHER ORDERED that Patent Owner's Motions to Exclude Evidence are *granted-in-part* with respect to Exhibits 1134–1137, 1140, 1141, 1143, 1147, 1234–1237, 1240, 1241, 1243, 1247, and 1264; *dismissed-in-part* with respect to Exhibits 1138, 1139, 1144–1146, 1148, 1153, 1156, 1158–1161, 1165, 1238, 1239, 1244–1246, 1248, 1256, 1260, and 1265; and *denied-in-part* with respect to Exhibits 1142, 1150, 1151, 1152, and 1250;

FURTHER ORDERED that Exhibits 1134–1137, 1140, 1141, 1143, 1147, 1234–1237, 1240, 1241, 1243, 1247, and 1264 shall be expunged; and

FURTHER ORDERED that, because this is a Final Written Decision, parties to the proceeding seeking judicial review of the decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

PETITIONER:

Bob Steinberg
Bob.steinberg@lw.com

Jonathan Link
Jonathan.link@lw.com

Giri Pathmanaban
Giri.pathmanaban@lw.com

PATENT OWNER:

David Hadden
Dhadden-ptab@fenwick.com