

UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE PATENT TRIAL AND APPEAL BOARD

---

ACTIFIO, INC.,  
Petitioner,

v.

DELPHIX CORP.,  
Patent Owner.

---

Case IPR2015-00100  
Patent 8,566,361 B2

---

Before KARL D. EASTHOM, JENNIFER S. BISK, and  
PATRICK R. SCANLON, *Administrative Patent Judges*.

SCANLON, *Administrative Patent Judge*.

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

## I. INTRODUCTION

Petitioner, Actifio, Inc., filed a Petition (Paper 1, “Pet.”) requesting an *inter partes* review of claims 1–6, 8, 14, 16–19, 24, and 25 of U.S. Patent No. 8,566,361 B2 (Ex. 1001, “the ’361 patent”) pursuant to 35 U.S.C. §§ 311–319. Patent Owner, Delphix Corp., filed a Preliminary Response (Paper 6, “Prelim. Resp.”). On May 14, 2015, we instituted an *inter partes* review as to all challenged claims (Paper 7, “Inst. Dec.”).

After institution, Patent Owner filed a Patent Owner Response (Paper 17, “PO Resp.”), and Petitioner filed a Reply (Paper 25, “Reply”). Petitioner relies on the Declaration of Erez Zadok (Ex. 1016) and the Supplemental Declaration of Erez Zadok (Ex. 1070) in support of its contentions, and Patent Owner relies on the Declaration of Prashant Shenoy, Ph.D. (Ex. 2016) in support of its contentions.

An oral hearing was held on February 2, 2016. A transcript of the hearing is included in the record. Paper 58 (“Tr.”).

We have jurisdiction under 35 U.S.C. § 6(b). This Final Written Decision is issued pursuant to 35 U.S.C. § 318(a) and 37 C.F.R. § 42.73. For the reasons that follow, we determine that Petitioner has shown by a preponderance of the evidence that claims 1–6, 8, 14, 16–19, 24, and 25 of the ’361 patent are unpatentable.

## II. BACKGROUND

### A. *Related Matters*

The parties indicate that the ’361 patent is at issue in *Delphix Corp. v. Actifio, Inc.*, No. 5:13-cv-04613-BLF (N.D. Cal.). Pet. 2; Paper 4, 2. The

'361 patent is also the subject of another petition for *inter partes* review filed by Petitioner—IPR2015-00108. Pet. 2; Paper 4, 2.

*B. The '361 Patent (Ex. 1001)*

The '361 patent describes systems and methods for managing databases and lifecycle workflows based on databases. Ex. 1001, 1:12–14. More specifically, the '361 patent involves creating one or more virtual databases based on a production database or another virtual database at a particular point in time. *Id.* at 5:64–66. Virtual databases are created “using storage level snapshots of production databases or clones of production databases instead of a live production database.” *Id.* at 6:16–19. “A virtual database created for a point in time is stored as a set of files that contain the information of the database as available at that point in time.” *Id.* at 6:31–34.

Figure 1 of the '361 patent is reproduced below.

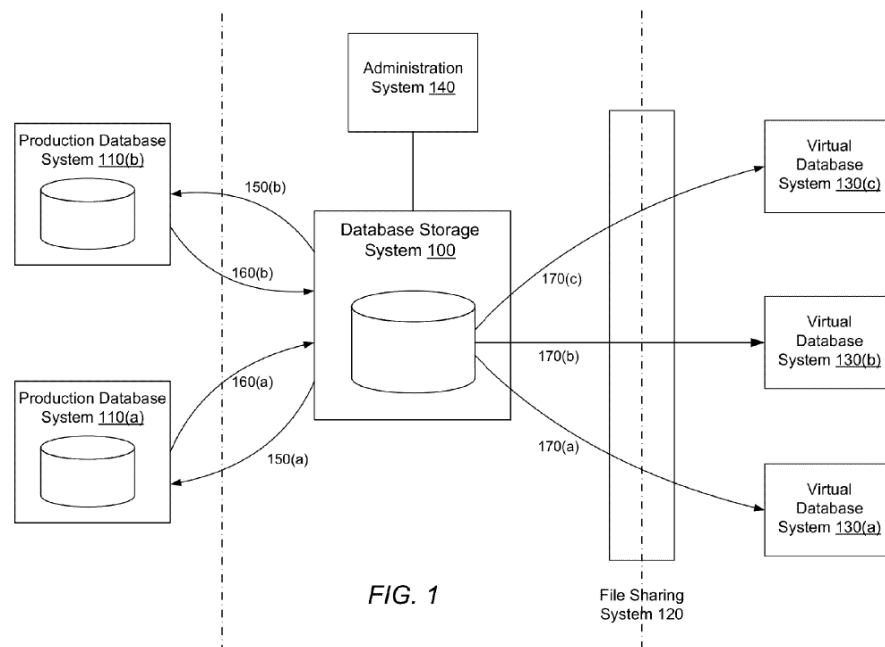


Figure 1 illustrates how information may be copied from production database systems 110 to database storage system 100. *Id.* at 7:12–15. To create a

virtual database, database storage system 100 creates files representing the corresponding information from a production database system at a given point in time. *Id.* at 7:50–52. More particularly, database storage system 100 retrieves and stores information from production database systems 110. *Id.* at 9:38–39. Database storage system 100 then exposes the files to virtual database system 130 using file sharing system 120. *Id.* at 7:53–55. Virtual database system 130 includes database server 230 and operating system 240. *Id.* at 7:60–8:6, Fig 2(a).

*C. Illustrative Claim*

Of the challenged claims in the '361 patent, claims 1, 14, 17, and 24 are independent. Claim 1 is illustrative of the claims at issue:

1. A method for replicating a database, the method comprising:
  - linking a source database, wherein linking the source database comprises receiving information identifying the source database;
  - loading the source database at multiple points in time, wherein the loading comprises:
    - receiving database blocks for different point-in-time copies of the source database, and
    - storing the database blocks in a first storage system;
  - replicating the database blocks of the source database from the first storage system to a second storage system; and
  - provisioning a virtual database (VDB) from the second storage system to a system running a database server, wherein provisioning comprises:
    - creating a set of files linked to the stored database blocks on the second storage system, and

mounting the set of files to the system allowing the database server running on the system to access the set of files.

Ex. 1001, 35:39–58.

*D. The Prior Art*

Petitioner relies on the following prior art:

1. JAWAHAR LAL, ROGER SANDERS & JEREMY BRUMER, DB2: CLONING A DATABASE USING NETAPP FLEXCLONE™ TECHNOLOGY, TR-3460 (2006) (“Sanders”) (Ex. 1004);
2. John K. Edwards et al., *FlexVol: Flexible, Efficient File Volume Virtualization in WAFL*, 2008 PROC. USENIX ANN. TECHNICAL CONF. 129 (“Edwards”) (Ex. 1005); and
3. DARRIN CHAPMAN, MIKE FEDERWISCH, & CHUCK DUFRESNE, SNAPMIRROR® BEST PRACTICES GUIDE, TR-3446 (2006) (“Chapman”) (Ex. 1006).

*E. Instituted Ground of Unpatentability*

We instituted the instant *inter partes* review on the following ground of unpatentability:

| References                    | Basis | Claims Challenged             |
|-------------------------------|-------|-------------------------------|
| Sanders, Edwards, and Chapman | § 103 | 1–6, 8, 14, 16–19, 24, and 25 |

### III. ANALYSIS

*A. Printed Publication—Sanders and Chapman*

Patent Owner contests that Sanders and Chapman are prior art “printed publications” 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). Public accessibility is a key question in determining whether a document is a printed publication and is determined on a case-by-case basis. *Suffolk Techs.*, 752 F.3d at 1364. To qualify as a printed publication, a document “must have been sufficiently accessible to the public interested in the art.” *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 and Chapman are 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 and Chapman being publicly accessible and prior art. *Id.* at 3–4. That position, however, is not informed by 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 challenge that included Sanders and Chapman.

Inst. Dec. 28; *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 Edwards, Sanders, and Chapman “were *published* years apart.” Prelim. Resp. 56, 58 (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 and Chapman, 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. PO Resp. 3–4. These declarations include declarations provided by Louis Hernandez (Ex. 1019) and Joseph Ortiz (Ex. 1027) in response to objections by Patent Owner<sup>1</sup> and a Supplemental Declaration by Mr. Hernandez (Ex. 1046) filed with its Reply. Our rules authorize serving supplemental evidence in response to an objection. 37 C.F.R. § 42.64(b)(2). Patent Owner lacks a foundation to complain that

---

<sup>1</sup> Exhibit 1027 is expunged at Petitioner’s request. We do not further discuss the Exhibit. We address Patent Owner’s motions to exclude these documents in a separate section, below.

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 and Chapman are not printed publications.

Turning to the substance of Exhibit 1019, 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. 1019 ¶¶ 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. "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. "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 Chapman, and that they were published during his tenure at NetApp. *Id.* ¶¶ 7, 9, 11, 17.

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

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. Reply 6–7 (citing Ex. 1046). Mr. Hernandez testifies that he uses the terms “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. 1046 ¶ 5. According to Mr. Hernandez, 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 reflected features for specific versions of NetApp’s products or if a document was outdated or updated to reflect more current features. *Id.* ¶ 10. Further, Petitioner provides evidence that by 2007 there were more than 94,000 NetApp systems deployed and the company had thousands of customers in 138 countries. Reply 6; Ex. 1055, 3.<sup>2</sup>

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 39)

---

<sup>2</sup> We find that Exhibit 1055, a NetApp Form 10-K SEC filing, was properly submitted by Petitioner as evidence in rebuttal to Patent Owner’s public accessibility challenge in its Response.

and Petitioner's Response (Paper 45), insofar as they relate to public accessibility of Sanders and Chapman. 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 and Chapman 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 and Chapman 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 Software Corp. v. Symantec Corp.*, Case IPR2014-00089, slip op. at 14 (PTAB Apr. 25, 2014) (Paper 9).

Finally, Petitioner also submits a declaration it says was produced in response to Patent Owner's evidentiary objections. Reply 4. Petitioner provided the declaration from the office manager of the Internet Archive (Wayback Machine). *Id.* (citing Ex. 1022). Sanders is dated April 30, 2006 and is designated "TR-3460" (or Technical Report 3460). Ex. 1004, 1. Chapman is dated April 2006 and is designated "TR-3446" (or Technical Report 3446). Ex. 1006, 1. According to the testimony regarding how the Internet Archive works (Ex. 1022 ¶¶ 3–5), we find the evidence indicates that Sanders and Chapman were available on NetApp's commercial website on, or at least as early as, November 22, 2006. Ex. 1022, 148, 231. Exhibit 1022 indicates that Sanders and Chapman were, thus, "sufficiently accessible to the public interested in the art." *See In re 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)). Petitioner asserts that the level of ordinary skill is not disputed in this case and that “there is no difference between customers and potential customers of NetApp and a [person of ordinary skill in the art].” Reply 6. Mr. Hernandez testifies that NetApp’s customers and potential customers were entities or individuals who were “interested in data storage technology, data management technology, database storage and management technology, and related technologies.” Ex. 1046 ¶ 3. As discussed above, Mr. Hernandez also testifies NetApp had more than two hundred Systems Engineers and other sales personnel during the period of 2004–2009 who distributed NetApp’s technical documents (including Sanders and Chapman) to “thousands” of customers and potential customers. *Id.* ¶ 4. Hence, the record supports Petitioner’s contention that Sanders and Chapman were distributed to persons interested and ordinarily skilled in the subject matter of database technology at the time of its publication. *See* Reply 6–7.

In view of the foregoing considerations, we find that Petitioner has established, by a preponderance of the evidence, that Sanders and Chapman were sufficiently disseminated to persons of ordinary skill interested in database technology to be deemed “publicly accessible” at least more than one year before October 21, 2009, the earliest possible priority date of the ’361 patent. *See* Ex. 1001, at [63]. Therefore, on this record, we determine Sanders and Chapman qualify as prior art printed publications under 35 U.S.C. § 102(b).

*B. Claim Construction*

We interpret claims of an unexpired patent using the broadest reasonable construction in light of the specification of the patent in which they appear. *See* 37 C.F.R. § 42.100(b). On this record and for purposes of this Decision, we determine that only the claim terms addressed below require express construction.

*1. “database block”*

In the Institution Decision, we preliminarily construed the term “database block,” as “a unit of data used by a database.” Inst. Dec. 11. In its Response, Patent Owner asserts that this definition is impermissibly broad. PO Resp. 18–28. Patent Owner’s proposed construction is “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.*” *Id.* at 18 (emphasis added). Petitioner, on the other hand, agrees with our conclusion in the Decision to Institute. Reply 9–13.

*a) Metadata Associated with the Database Block*

The main dispute between the parties centers on whether a database block must necessarily include metadata. Patent Owner asserts that it does (PO Resp. 19–23), and Petitioner disagrees (Reply 9–13). We agree with Petitioner.

We begin our analysis by considering the language of the claims themselves. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1314, 1315 (Fed. Cir. 2005) (*en banc*). The term “metadata” is not recited in any of the claims of

the '361 patent. U.S. Patent No. 8,150,808 B2 (“the '808 patent”),<sup>3</sup> however, includes two claims that recite “metadata”—dependent claims 32 and 33, which are not challenged in any proceeding, of which we are aware, currently before the Board. These claims depend indirectly from claim 1 of the '808 patent and expressly recite “metadata of database blocks.” Thus, had the patentees intended to limit “database blocks” recited in the claims of the '361 patent to require metadata, they demonstratively could have done so by explicitly modifying the disputed term with “metadata,” but did not.

Moving to the specification, Patent Owner asserts that the following passage in the Summary section of the '808 patent 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. 18 (quoting Ex. 2002, 2:7–12).

The first phrase in the cited passage above explicitly defines the term, by stating “[a] database block is a unit of data used by a database.”<sup>4</sup> Ex. 2002, 2:7–8. Although the sentence following shortly thereafter states that a database block “stores metadata,” that sentence by itself is insufficient to limit the disputed term by requiring the unclaimed “metadata” feature because it

---

<sup>3</sup> Patent Owner describes the '808 patent as being related to the '361 patent. Prelim. Resp. 28.

<sup>4</sup> The first sentence also states a database block comprises “a specific number of bytes stored in the storage.” For the reasons discussed below, we find this addition is not part of the explicit definition but, rather, represents embodiments within the defined term. Patent Owner also fails to explain how to interpret this particular phrase or how it presents a material issue related to the prior art.

does not state unambiguously that all “database blocks” must include metadata. *See Liebel-Flarsheim Co. v. Medrad, Inc.*, 358 F.3d 898, 906 (Fed. Cir. 2004) (construing a claim term broadly because “[n]o statement in the written description [ ] constitute[d] a limitation on the scope of the invention”) (alterations in original) (quoting *Brookhill-Wilk 1, LLC v. Intuitive Surgical, Inc.*, 334 F.3d 1294, 1301 (Fed. Cir. 2003)). Further, the cited passage also does not exclude the possibility of some database blocks not having any metadata. *See id.* at 908 (passages in the Summary of the Invention 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”). More importantly, the rest of the ’361 patent is consistent with a definition of database blocks that does not require metadata in all database blocks.

Patent Owner asserts, citing certain portions of the ’361 patent and the Declaration of Prashant Shenoy, Ph.D. (Ex. 2016), that if a database block does not include metadata, the disclosed system would not work as described. PO Resp. 19–20. 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, which is “one of the main functions” of the claimed system (*id.* at 19 (citing Ex. 1001, 7:31–35, 41–44, 14:43–46; Ex. 2016 ¶¶ 86–87)) essential to achieving “a main purpose” of the invention—“to efficiently provide virtual databases . . . without proliferating redundant copies of database data” (*id.* at 19–20 (citing Ex. 2016 ¶ 87)). 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 20 (citing Ex. 1001, 15:27–31).

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.” (quoting *E-Pass Techs., Inc. v. 3Com Corp.*, 343 F.3d 1364, 1370 (Fed Cir. 2003))).

Moreover, Patent Owner focuses its arguments on the streaming embodiment, however, Patent Owner does not show that the file sharing embodiment requires this approach, 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.*, Ex. 1001, 14:10–14 (“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” (emphases added)),<sup>5</sup> 7:12–18 (“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)). When file sharing, as opposed to streaming, is used, the database file on the production system can be accessed and copied directly by “mount[ing]

---

<sup>5</sup> The phrase “metadata *associated* with database blocks” implies any metadata need not be *in* the database blocks.

the production DB data store” on the database storage system without packing and unpacking the database blocks of the database file into and out of data streams. *Id.* at 8:58–65.

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, 14, 17, and 24 recite receiving database blocks for different point-in-time copies of the source database and storing the database blocks in a storage system. But the claims do not say anything about a particular method of receiving point-in-time copies, whether by streaming or by file sharing. Hence, the claims are not limited to either embodiment, and, therefore, the streaming embodiment cited by Patent Owner does not limit the claims.

Further, even looking at the streaming embodiment, we are not persuaded that metadata is required by database blocks. Patent Owner points only to a single streaming embodiment (PO Resp. 20), in which 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. Ex. 1001, 14:43–60. 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 14:64–15:3 (“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 second 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, nothing in the specification 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).

Therefore, in view of the entire disclosure of the ’361 patent and the plain language of the claims, we find that the intrinsic record before us does not justify limiting the term “database block” by reading in the “metadata” limitation not found in the claims. *See Liebel-Flarsheim Co.*, 358 F.3d at 908.

Patent Owner also argues additional evidence supports its proposed construction requiring data blocks to include metadata. For example, citing the testimony of Dr. Shenoy, Patent Owner asserts that all database management systems mentioned in the ’361 patent, such as Oracle and IBM DB2 (Ex. 1001, 6:5–9), require metadata in database blocks. PO Resp. 21–22 (citing Ex. 2016 ¶¶ 41–46, 84–85). In the paragraphs cited by Patent Owner, Dr. Shenoy discusses various documents describing the database systems listed in the ’361 patent, including Oracle, Sybase, Microsoft SQL Server, and IBM DB2, and testifies that these database systems all require metadata in

database blocks.<sup>6</sup> Ex. 2016 ¶¶ 43–46, 84–85 (citing testimony from Petitioner’s declarant, Dr. Zadok in another related case). Relying upon the testimony of Dr. Shenoy, Patent Owner also asserts that “[i]t is well known to one skilled in the art that a database block necessarily includes metadata.” PO Resp. 21 (citing Ex. 2016 ¶¶ 84–85); *see also id.* at 19 (“[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. 2016 ¶¶ 41–46, 84)). In his Declaration, Dr. Shenoy discusses a treatise on database systems (Ex. 2019, “Database Systems”) and testifies that it is generally understood that a database block or a page will include metadata. Ex. 2016 ¶¶ 40–42 (citing Ex. 2019, 29, 31).

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 ’361 patent—is properly characterized as extrinsic evidence. Such extrinsic evidence is “less significant than the intrinsic record in determining the legally operative meaning of claim language.” *Phillips*, 415 F.3d at 1317 (quoting *C.R. Bard, Inc. v. U.S. Surgical Corp.*, 388 F.3d 858, 862 (Fed. Cir. 2004) (internal quotation marks omitted)).

Petitioner argues that Patent Owner does not provide any evidence that database blocks require metadata in database systems other than relational database systems. *See* Reply 10–11. Petitioner correctly points out that the ’361 patent specification states that the disclosed invention “can be used for

---

<sup>6</sup> However, neither Patent Owner’s brief nor Dr. Shenoy’s testimony discuss the MYSQL database system mentioned in the ’361 patent.

*any database.*” *Id.* at 10 (quoting Ex. 1001, 6:15–16). Further, 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. 2002, 5:8–15 (emphasis added). Thus, we agree with Petitioner that the ’361 patent does not limit its disclosed invention to relational database technology.

Similarly, the portion of the Database Systems treatise relied upon by Dr. Shenoy appears to describe features of a *relational* database system, not characteristics common to all databases in general. *See, e.g.*, Ex. 2019, 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)).

Relying on the testimony of Dr. Zadok submitted with its Reply (Ex. 1070), Petitioner asserts that, in other type of database systems, such as Google’s BigTable database, metadata is stored separately from the database blocks.<sup>7</sup> Reply 11 (citing Ex. 1047, 4; Ex. 1070 ¶ 24). Petitioner also argues,

---

<sup>7</sup> 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* Reply 10–11 (citing Ex. 2016 ¶ 33). In addition to the disclosure that “the techniques disclosed can be used for any database,” (Ex. 1001, 6:14–15) the ’808 patent provides another broad description—“[a] database comprises data stored in a computer for use by computer implemented applications” (Ex. 2002, 4:67–5:1). Under either description or

citing the testimony of Dr. Zadok, that a “flat file database,” such as a comma separated value (CSV) file used by spreadsheet applications, does not have metadata in its blocks. Reply 11 (citing Ex. 1070 ¶ 24). Therefore, Petitioner asserts that the ordinary meaning of database block does not require including metadata. *See id.*; Ex. 1070 ¶ 22.

We agree with Petitioner that all extrinsic evidence Patent Owner relies upon to argue ordinary meaning appears to be directed to relational databases. Because the challenged claims plainly recite “database,” and, therefore, are not limited to relational databases, Patent Owner’s evidence, even assuming it shows relational databases all 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.”( quoting *Key Pharm. v. Hercon Labs. Corp.*, 161 F.3d 709, 716 (Fed. Cir. 1998)) (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

---

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, flatfiles, and spreadsheet databases, and “any database.”

the field of database systems that requires metadata to be necessarily included in a database block.

b) “*Application Level*” v. “*Storage Level*”

Patent Owner also argues, citing the passages in the ’808 patent describing the streaming embodiment, that a “database block” must include metadata because “the ’361 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. 28 (emphasis added); *see also id.* at 16–18 (making the same argument) (citing Ex. 1001, 14:43–60; Ex. 2016 ¶¶ 86, 87, 89, 90). Patent Owner asserts, therefore, a “database block” cannot be a “*storage level* unit of data.” *Id.* at 25 (emphasis added). The passages of the ’808 patent cited by Patent Owner, however, describe solely the streaming embodiment and, therefore, do not limit the challenged claims as discussed above.

Patent Owner also asserts that the system of the ’361 patent “operates at the application level” because the system uses APIs (application program interfaces) to copy database blocks. *Id.* at 16 (citing Ex. 1001, 8:48–53; Ex. 2016 ¶¶ 69, 88). Petitioner argues that the ’361 patent never mentions “application level,” teaching instead that “the system ‘can create a virtual database using *storage level snapshots* of production databases’” and “retriev[ing] ‘database blocks from *storage level snapshots*.’” Reply 7–8 (quoting Ex. 1001, 6:17–18, 9:35) (emphases added by Petitioner). We agree with Petitioner’s argument and further note that the ’361 patent also discloses copying of database blocks from *storage level snapshots* of a database. *See* Ex. 1001, 7:12–18 (“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 ’361 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. 25–28 (citing Ex. 2016 ¶¶ 27–33). In particular, Patent Owner asserts that an Oracle document (Ex. 2004, “Oracle Database Concepts 11g Release 2 (11.2)”) explains this “well-known distinction.” *Id.* at 25–26 (citing Ex. 2004, 250, Fig. 12-5). Patent Owner further argues “neither are file system blocks accessible to a database nor are database blocks accessible to the file system.” *Id.* at 27 (citing Ex. 2016 ¶¶ 95–99). Therefore, Patent Owner argues a construction of the term “database block” that encompasses a storage unit block is incorrect. *Id.* at 28.

As a threshold matter, we note that the Oracle document is dated May, 2014, which is several years later than the filing date of the ’361 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 ’361 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” or a “storage 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. Inc. v. Stamps.com Inc.*, 582 F.3d 1341, 1348 (Fed. Cir. 2009); *Phillips*, 415 F.3d at 1318.

c) *A Specific Number of Bytes Stored in the Storage*

In the Institution Decision, we found that database blocks of the ’361 patent need not 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 the storage.” Inst. Dec. 9–10. Patent Owner continues to argue that this is a requirement of the proper construction of “database blocks.” PO Resp. 18.

Patent Owner does not explain how to interpret “a specific number of bytes stored in the 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 any number of different sizes independent of the size of the file system data blocks which may ultimately store the database data.” *Id.* at 27 (citing Ex. 2016 ¶ 96). 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 27 (citing Ex. 1007, 28).

According to Patent Owner’s arguments, a database block may be “any number of different sizes.” Therefore, the database blocks of the ’361 patent need not have a constant specific number of bytes stored in storage.

d) “Database Block” v. “File”

Next, Patent Owner argues that the construction proposed by Petitioner (Pet. 9) and preliminarily adopted in our Institution Decision (Inst. Dec. 11)—“a unit of data used by a database”—is improper because it would equate a “database block” with a “file.” PO Resp. 24–25. Citing dictionaries, Patent Owner argues that the ordinary meaning of “file” is “a unit of data.” *Id.* at 24 (citing Ex. 2026, 3; Ex. 2027, 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.* at 24–25.

Petitioner argues, citing the deposition testimony of Patent Owner’s expert, Dr. Shenoy, that Patent Owner is incorrect because a file has a name associated with it but a database block, as construed by Petitioner, would not. Reply 13–14 (citing Ex. 1030, 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. 2026, 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. 2027, 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.

*e) 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, 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.” (quoting *Renishaw plc v. Marposs Societa’ per Azioni*, 158 F.3d 1243, 1250 (Fed. Cir. 1998))).

2. “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. 16. Petitioner agrees with this construction. Reply 14–18. Patent Owner, on the other hand, asserts that this definition reads out the “virtual” requirement. PO Resp. 31. Patent Owner, instead proposes that the term 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 28.

To analyze Patent Owner’s proposed construction, Patent Owner defines a virtual database as “a set of files” or “database 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) “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.

*a) Database Server Can Read and Write*

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. 32. 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.

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.” *Id.* Patent Owner’s argument is unpersuasive because the same objective can be accomplished without inserting the extraneous term by directly stating a virtual database must be “a set of database files.” In fact, the ’361 patent describes that a virtual database

comprises “database files.” *See, e.g.*, Ex. 1001, 6:19–22 (“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.” (emphases added)), 40–41 (“A virtual database may be created on a database server by creating the *database files . . .*” (emphasis added)). Patent Owner appears to argue this is insufficient because “[t]o be a database file, a database server must be able to read, write and understand its contents.” PO Resp. 32 (citing Ex. 2016 ¶ 109). We are not persuaded by this argument because it does not comport with the written description of the ’361 patent.

First, the ’361 patent provides a description of a “database” as follows: “[a] database comprises data stored in a computer for use by *computer implemented applications*.” Ex. 1001, 6:1–2 (emphasis added). Furthermore, the ’361 patent announces that the disclosed invention “can be used for *any database*.” *Id.* at 6:14–15. Hence, a database file may be used by an application or a program but not necessarily a “database server,” which is a server-side program in a client-server system or network. *See id.* at 6:2–4, 31:43–46.

Furthermore, the ’361 patent describes that various programs that are not a database server can read and write to a database or database file and understand its content. For example, the virtual database manager program at the database storage system can create, read, and write to read/write files representing a virtual database. *See* Ex. 1001, 7:3–5; Ex. 2002, 17:57–63, 18:27–31. Other programs of the database storage system, such as the point-in-time copy manager and the storage allocation manager, can create, read,

and write to database files based on database blocks received from the production system. *See* Ex. 1001, 10:46–63, 11:13–31.

Therefore, contrary to Patent Owner’s contention, the written description of the ’361 patent shows that a database does not require a database server reading and writing to the database. Patent Owner’s proposed construction would improperly import a particular embodiment from the specification by reading in the operation of a database server to impermissibly narrow the meaning of the term “virtual database.” As discussed above, features of particular embodiments may not be read into the claims “absent clear disclaimer in the specification.” *Am. Acad. of Sci. Tech Ctr.*, 367 F.3d at 1369.

On the other hand, the specification shows that the readable and writable characteristics are inherent attributes of a virtual database. *See, e.g.*, Ex. 1001, 5:64–6:1 (“the virtual databases can then be individually accessed and modified as desired”); Ex. 2002, 18:27–28 (“FIG. 10 indicates how storage efficient copies are made to create a *read/write file structure representing a VDB.*” (emphasis added)), 17:57–63 (“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 PIT copy.” (emphasis added)), 18:12–15 (“The virtual database manager [] sends . . . handles to the *read/write file structure* to the associated virtual database system 130.” (emphasis 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 definition 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.”

*b) 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 '361 patent to support the second proposed modifying phrase—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. 29 (quoting Ex. 1001, 6:19–22). Patent Owner asserts this sentence explicitly defines the use of the word “virtual” in the term “virtual database.” *Id.* However, the '361 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 '361 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. 2016 ¶¶ 23–26, 101). Petitioner asserts Patent Owner confuses “platform virtualization” with “database virtualization” or “virtualizing data.” *See* Reply 16–17. We agree with Petitioner that “platform virtualization” is unrelated to the concept of “virtual database” as claimed and described in the '361 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. 2016 ¶¶ 24–25 (citing Ex. 2003, 451–52, Fig. 24-7). Patent Owner’s discussion of 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.” Ex. 2016 ¶¶ 23–24 (emphasis added). The ’808 patent also describes that a “virtual machine” is “provided by *platform virtualization software*” or “server virtualization *software*.” Ex. 2002, 7:16–19 (emphases added). Further, the ’361 patent describes a Java Virtual Machine as part of production database system 110 (Ex. 1001, 9:13–15), which is separate from virtual database 220 stored on database storage system 100, as discussed above. Hence, Patent Owner’s discussion of the Java Virtual Machine (PO Resp. 32 n.9) 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. 2002, 6:49–51, 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 ’361 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. 2016 ¶ 69 (quoting Ex. 1001, 8:48–53). The cited portion of the ’361 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. 1001, 8:53–56 (“An example of a vendor interface module is the *program code* of a database server provided by vendor ORACLE that implements RMAN APIs.” (emphasis added)), 14:10–12 (“the production system library 385 includes *code* to analyze the structures of the files of the database stored in the data store 350” (emphasis added)).

Dr. Shenoy similarly testifies that “each database (130c)” depicted in Figure 1 of the ’361 patent is “virtual” (Ex. 2016 ¶ 102; *see also* PO Resp. 29 (citing Ex. 2016 ¶ 102)) when, in fact, Figure 1 shows the part identified by Patent Owner is Virtual Database System 130(c), *not* the virtual database, which is stored in a different system, namely, Database Storage System 100. As described in Figure 2a of the ’361 patent, virtual database system 130 is separate from and external to *virtual database* 220 stored on database storage system 100. Further, the ’361 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. 1001, 13:3–13 (“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*. Reply 17–18. We agree with Petitioner, but our analysis underscores a related salient point, which is that 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 ’361 patent, a database system generally comprises a database, which is data stored in storage, and database software, such as a database server or other program that accesses the database. *See* Ex. 1001, 6:1–4, 8:36–65. 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. 2016 ¶ 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 importing the function or operation 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. If patentee intended to claim the function or 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. 31 (citing Inst. Dec. 16). Patent Owner’s argument is unpersuasive because our refusal to read the functions and 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 ’361 patent.

Turning to the claim language and the written description, the challenged independent claims recite, with emphasis added, provisioning a first “virtual database” including creating “a set of files *linked* to the stored database blocks.” Similarly, the Specification describes as follows:

A virtual database may be created on a database server by creating the database files for the production database corresponding to the state of the production database at a

previous point in time, as required for the database server. The files corresponding to the virtual database are made available to the database server using a file sharing mechanism, which links the virtual database to the appropriate database blocks stored on the storage system.

Ex. 1001, 6:40–47 (emphasis added); *see also* Ex. 2002, 2:24–27 (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 '361 patent depicts “how . . . to create a read/write file structure representing a VDB.” Ex. 1001, 19:27–28. 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 19:47–49 (emphasis added). The '361 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 19:49–55 (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 a copy of the original block of data F11 at another physical location:

For example, if the virtual database system 130 writes to the block V11, space is allocated and block F11 copied to the

allocated block. Hence the original copy of the block F11 is maintained as a read only copy and the virtual database system 130 is allowed to write to a copy of the appropriate database block created specifically for the virtual database system 130.

*Id.* at 20:51–57. The '361 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 20:44–48. In addition, “[a] database block that is not written to by virtual database systems 130 may be shared by several virtual database systems without being copied for a specific virtual database systems 130.” *Id.* at 20:64–67.

Thus, the '361 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.

Figure 12 also shows how database blocks may be shared by file structures created for several virtual databases. Ex. 1001, 19:63–65. File structures 1050 and 1150 include blocks V11–V14 and U11–U14, respectively. *Id.* at 19:65–20:8. Each of these blocks is implemented as a pointer to the actual database block storing the relevant data. *Id.* Thus, several blocks in the virtual database may share the same database blocks—V11 and U11 both point to the same copy of F11. Thus, the '361 patent indicates that the database files are “virtual” in the sense that they create the illusion of storing data, but the system actually stores the data elsewhere in another physical location (F11) specified by the pointing blocks such as V11

and U11. The '361 patent also describes the procedure for writing to a shared database block, such as V11: “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 20:51–53.

Based on the disclosure in the '361 patent discussed above, “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 '361 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” in part, because “[c]ommon database blocks are shared between different point-in-time virtual copies. This is imperceptible to the . . . user.” Ex. 2016 ¶ 63. Similarly, Dr. Shenoy testifies 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.* ¶ 107 (emphasis added). Nonetheless, Patent Owner argues that the preliminary construction is erroneous because database files of any database, including a source database, can be “mapped to physical addresses for stored database blocks.” PO Resp. 31 (citing Ex. 2016 ¶¶ 109–10). Apparent inconsistencies in Dr. Shenoy’s testimony aside, Patent Owner’s argument is unpersuasive because the '361

patent expressly describes that a source database can be a virtual database. *See* Ex. 1001, 20:17–18. 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 since an ordinary, non-virtual database consists of its own database blocks and would *not* be created by mapping or pointing to stored blocks of a different database. This tracks the claim language of the challenged claims, which define how to create a VDB, which in turn, tracks Dr. Shenoy’s testimony. *See* Ex. 2016 ¶¶ 63, 107.

Furthermore, although not argued by the parties, it is not necessary to require that the mapped database blocks are associated with a point-in-time copy of a database because the challenged independent claims expressly recite virtual database files wherein at least some of the database blocks are associated with multiple “point-in-time copies of the source database.” *See Digital-Vending Servs. Int’l, LLC v. Univ. of Phoenix*, 672 F.3d 1270, 1274–75 (Fed. Cir. 2012). 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.”

*C. Asserted Ground Based on Sanders, Edwards, and Chapman*

Petitioner challenges claims 1–6, 8, 14, 16–19, 24, and 25 as obvious under 35 U.S.C. § 103 over Sanders, Edwards, and Chapman. Pet. 3, 28–59. To support this assertion, Petitioner relies on the Declaration of Dr. Erez Zadok (Ex. 1016).

1. *Overview of Sanders*

Sanders describes a system “to clone a DB2 database quickly and easily.” Ex. 1004, 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 snapshot created for the clone; or applications can point directly to the clone database.” *Id.* at 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:

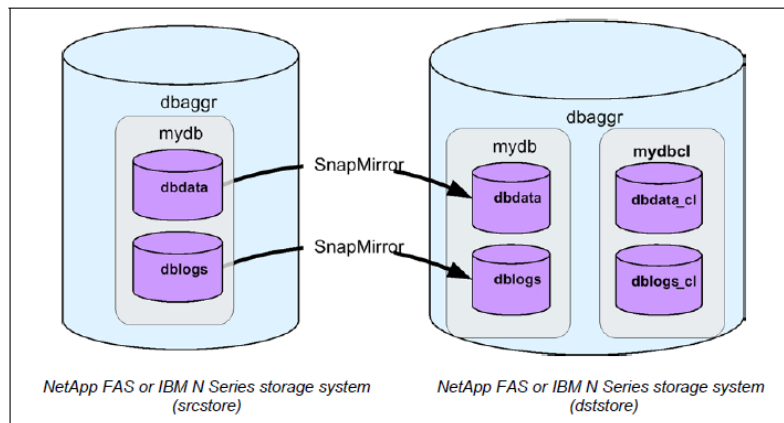


Figure 3) Clone database volumes on the second NetApp FAS or IBM N Series storage system

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].”).

## 2. Overview of Edwards

Edwards describes creating virtual instances of WAFL (write anywhere file layout) file systems; read-only versions, called “FlexVol” volumes (Ex. 1005, 14)<sup>8</sup> and writeable versions, called “FlexClone” volumes (*id.* at 14–15). A FlexVol volume “break[s] this tight bond between volumes and their underlying storage” by hiding a file system, which spans a pool of storage, and creating externally visible volumes inside the files of the hidden file system. *Id.* at 11. This “introduces a level of indirection, or virtualization,

<sup>8</sup> When referencing Exhibit 1005, we refer to the pagination inserted by Petitioner.

between the logical storage space used by a volume and the physical storage space.” *Id.*

Edwards discloses that writable versions of the FlexVol volumes, FlexClones, are particularly useful in database environments—“for example, it is often desirable to make writable copies of a production database for development or test purposes.” *Id.* at 14. To create a FlexClone volume, WAFL creates a copy of vol\_info block of the Snapshot copy on which the clone is based. *Id.* at 15. The FlexClone volume, thus, inherits pointers to the complete file system stored in the original Snapshot copy. *Id.*

### 3. *Overview of Chapman*

Chapman describes NetApp’s SnapMirror technology and is “intended to serve as a deployment guide for architecting and deploying SnapMirror in a customer environment.” Ex. 1006, 1. As one of several features, Chapman discloses “cascading” as a “variation on the basic SnapMirror deployment and function [that] involves mirroring from established mirrors to more SnapMirror destinations.” *Id.* at 22.

### 4. *Independent claims 1, 14, 17, and 24*

Regarding independent claim 1, Petitioner argues that “Sanders discloses a process for replicating a database.” Pet. 28 (citing Ex 1004, 3; Ex. 1016 ¶ 114). In particular, Petitioner asserts that Sanders discloses receiving information identifying a source database—and, thus, linking the source database—because the reference discloses “creating a snapshot copy of a source volume containing an IBM DB2 database on a source storage system and then mirroring the copy from the source volume to a destination volume on a first storage system.” *Id.* (citing Ex 1004, 22–23; Ex. 1016 ¶ 116).

According to Petitioner, Sanders discloses loading a source database at multiple points in time (*id.* at 29 (citing Ex. 1016 ¶ 118)), and Edwards discloses that the “data blocks” or “blocks” of Sanders are units of data used by a database (i.e., “database blocks”) (*id.* at 31 (citing Ex. 1005, 21)). Petitioner then asserts that “Sanders in view of Edwards discloses receiving and storing database blocks for different point-in-time copies of a source database in a first storage system.” *Id.* According to Petitioner, one of ordinary skill in the art “would have known to combine the NetApp Data ONTAP technologies described in Sanders with the teachings of Edwards . . . in order to discern the units of data used by a database in WAFL.” *Id.* at 31–32 (citing Ex. 1016 ¶ 124).

Petitioner asserts that Sanders discloses using “NetApp’s SnapMirror technology for data replication” and Chapman discloses “the use of data cascading, which is a ‘variation on the basic SnapMirror deployment and function [that] involves mirroring from established mirrors *to more SnapMirror destinations.*’” *Id.* at 32 (quoting Ex. 1006, 22) (citing Ex. 1004, 22, 27; Ex. 1016 ¶ 126). Petitioner then asserts:

Given Sanders’s disclosure that a “SnapMirror source and its corresponding destination can reside on [...] two separate storage systems that are miles apart” (Sanders at 3), a POSITA would have been motivated to incorporate Chapman’s teachings that cascading “make[s] a uniform set of data available on a read-only basis to users from various locations throughout a network” (Chapman at 22) and can thus be “very useful for data distribution.” *Id.* at 11. Thus, Sanders in view of Chapman discloses [the replicating database blocks] limitation. Zadok Decl. at ¶ 129.

*Id.* at 33 (first and second alterations in original).

Petitioner asserts that Sanders discloses making a “virtual database available to a database server running on a storage system through the use of FlexClone to create clones of databases on FlexVol volumes, followed by the mounting of those clones to a database server.” *Id.* at 34 (citing Ex. 1004, 3, 28; Ex. 1016 ¶ 130). In addition, Petitioner argues that Chapman discloses that “any volume that resides on a cascaded, second destination storage system can be cloned.” *Id.* (citing Ex. 1016 ¶ 131). Petitioner concludes the combination of Sanders and Chapman discloses the limitation of provisioning a virtual database. *Id.* at 35 (citing Ex. 1016 ¶ 133).

Petitioner argues that Sanders discloses creating a database clone that “is a frozen image of the database file system at the time of the clone creation.” *Id.* (quoting Ex. 1004, 6 (emphasis omitted)) (citing Ex. 1016 ¶ 135). Petitioner also argues that Edwards discloses “further detail on creating a set of files linked to the stored database blocks on the storage system, through the creation of a FlexClone volume.” *Id.* at 35–36 (citing Ex. 1016 ¶ 135). Petitioner contends that one of ordinary skill in the art “would have looked to Edwards to understand the specifics of how FlexClone creates files linked to stored database blocks on the storage system.” *Id.* at 37–38 (citing Ex. 1016 ¶ 140).

Finally, Petitioner argues that “Sanders discloses a command to mount cloned NetApp volumes” that identifies the name assigned to the mount location. *Id.* at 38 (citing Ex. 1004, 29; Ex. 1016 ¶ 141). According to Petitioner, Sanders thus discloses the mounting a set of files limitation because “[b]y identifying a mount location in the storage system, the set of files in the cloned volume can be made accessible to the system over a file

sharing system (NFS), allowing a database server running on the system to access the clone volumes.” *Id.* (citing Ex. 1004, 28; Ex. 1016 ¶ 142).

Independent claim 14 is directed to a “method for backup of source databases using a virtual database system.” Ex. 1001, 37:17–18. Petitioner asserts that the limitations of claim 14 are similar to the limitations of claim 1, and that claim 14 is rendered obvious by the combination of Sanders, Edwards, and Chapman for essentially the same reasons set forth in connection with claim 1. Pet. 46–51.

Independent claim 17 is directed to a “computer program product having a non-transitory computer-readable storage medium storing computer-executable code for replicating a database.” Ex. 1001, 37:54–56. Petitioner argues that the computer executable code of claim 17 comprises instructions to “perform steps essentially identical to those recited in Claim 1.” Pet. 55. Thus, Petitioner asserts that, “for the reasons discussed above in connection with Claim 1, Sanders, in view of Edwards and Chapman, discloses and renders obvious the computer program product elements of Claim 17.” *Id.*

Independent claim 24 is directed to a “computer program product having a non-transitory computer-readable storage medium storing computer-executable code for performing backup of source databases using a virtual database system.” Ex. 1001, 39:9–12. Petitioner argues that the computer executable code of claim 24 comprises instructions to perform steps that are not materially distinguished from the steps of Claim 14. Pet. 58. Thus, Petitioner asserts that, “for the reasons discussed above in connection with Claim 14, Sanders, in view of Edwards and Chapman, discloses and renders obvious the computer program product elements of Claim 24.” *Id.* (citing Ex. 1016 ¶ 200).

a) *“linking a source database, wherein linking the source database comprises receiving information identifying the source database”*

Patent Owner argues that Petitioner relies solely on Sanders’s disclosure of a SnapMirror initialize command as disclosing this linking limitation, but this command identifies a NetApp volume rather than a source database. PO Resp. 39–40 (citing Ex. 2016 ¶ 170). According to Patent Owner, “there is no necessary correspondence between a volume and a database,” and “an identification of a volume is not an identification of a database.” *Id.* at 40 (citing Ex. 2016 ¶ 172). Patent Owner also argues that “in addition to ‘receiving information identifying the source database,’” the claim language requires “actual ‘link[ing]’ of that source database.” *Id.* (alteration in original). Patent Owner contends that Sanders does not disclose such “link[ing].” *Id.* at 40–41.

In response, Petitioner argues that Sanders explicitly discloses a correspondence between a volume and a database. Reply 21. In particular, Petitioner asserts that Sanders’s SnapMirror initialize command identifies a source volume named “dbdata,” and “dbdata” is “used to store the production database table data.” *Id.* (citing Ex. 1004, 6, 22). Relying upon the testimony of Dr. Zadok, Petitioner contends that “[t]here is no reason for a [person of ordinary skill in the art] to create a NetApp volume with more than one database.” *Id.* (citing Ex. 1070 ¶ 58). Regarding Patent Owner’s “actual linking” argument, Petitioner argues that the claim language defines “linking” as “receiving information identifying the source database” and does not require “linking” to including anything more than receiving the information. *Id.* at 20–21 (citing Ex. 1001, 35:42–43; Ex. 1070, ¶ 57).

We find a preponderance of the evidence demonstrates that Sanders discloses the linking limitation. We are not persuaded that, as Patent Owner argues, an identification of a volume cannot be an identification of a database. Rather, we credit Dr. Zadok's testimony that "there is no reason for a [person of ordinary skill in the art] to think that a volume designated for and containing a database does not identify a database." Ex. 1070 ¶ 58. We also are persuaded by Petitioner's argument that the claim language does not require "linking" to include anything more than "receiving information identifying the source database." Thus, we are not persuaded by Patent Owner's argument that Sanders fails to disclose "actual linking."

b) *"database blocks of the source database"*

Patent Owner argues that the limitations of receiving "database blocks for different point-in-time copies of the source database" and storing "the database blocks in a storage system" are not rendered obvious by the combination of references asserted by Petitioner "under any proposed construction." PO Resp. 41.

Patent Owner first argues that the proposed combination of references does not render obvious the "database blocks" limitations even under Petitioner's proposed construction, which, as discussed above, we adopt. *Id.* at 41–46. According to Patent Owner, "Petitioner contends that WAFL data blocks [of Edwards] are the claimed 'database blocks.'" *Id.* at 42 (citing Pet. 31–32). Patent Owner argues that Edwards's WAFL data blocks are not equivalent to database blocks because they are not "used by a database." *Id.* Instead, Patent Owner asserts that no database has knowledge of the WAFL data blocks "existence, where they are located, or how to interact with them." *Id.* (citing Ex. 2016 ¶ 131).

Patent Owner’s argument, however, is premised on the previously discussed “application level” versus “storage level” distinction—the assumption that “[d]atabase servers only use database blocks which are data structures that can be accessed or used at the application level” and “only the database server,” but not the WAFL file system, “has knowledge of and manipulates the database block as an independent unit of data.” *Id.* at 42–43 (citing Ex. 2016 ¶¶ 132, 133, 142). As discussed above, we reject this premise as divorced from the written description of the ’361 patent.

Patent Owner next argues that the asserted references do not render obvious the “database blocks” limitations under Patent Owner’s proposed construction. *Id.* at 46–48. This argument is not persuasive, however, because we do not adopt Patent Owner’s proposed construction, as discussed above.

For the above reasons, we conclude that a preponderance of the evidence demonstrates that a person of ordinary skill would have found the claimed database blocks obvious over the data blocks disclosed by Edwards.

c) “*point-in-time copies of the source database*”

Patent Owner argues that Petitioner’s identification of the Snapshot copies disclosed by Sanders as the claimed point-in-time copies is wrong. *Id.* at 49. Patent Owner contends that “when a copy of a FlexVol volume is created by SnapMirror, it is the exact replica of the *current* state of a FlexVol volume” and “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.* at 50 (citing Ex. 2016 ¶¶ 151, 158–160, 164). Patent Owner further argues that “merely cloning the current state of the volume is fundamentally different than the invention of the ’361

patent,” and SnapMirror “cannot be used to accumulate any ‘database blocks’ reflecting the state of the source at different points-in-time.” *Id.* at 51 (citing Ex. 2016 ¶¶ 160–64).

In response, Petitioner argues that while Patent Owner “asserts that ‘a Snapshot copy must exist in both the source and destination volume to create a clone from the destination volume,’” Patent Owner does not explain how this statement could mean that database blocks for different point-in-time copies are not received and stored. Reply 20 (citing PO Resp. 50–51).

We are not persuaded by Patent Owner’s arguments. The crux of these arguments appears to be that, because a Snapshot copy allegedly is an exact copy of the current state of the volume, SnapMirror does not utilize database blocks *from different points in time* to create a Snapshot copy. The claim language, however, does not require database blocks for different point-in-time copies to be utilized in this manner. Rather, the claim language merely requires receiving and storing database blocks for different point-in-time copies. We agree with Petitioner that the combined teachings of Sanders and Edwards disclose receiving and storing database blocks for different point-in-time copies. Sanders discloses that “[t]he SnapMirror initialize command creates a Snapshot copy of the source volume and transfers data from the source to the destination volume,” such that the transferred data is received and stored at the destination volume. Ex. 1004, 22. We also credit Dr. Zadok’s testimony that a Snapshot copy is a point-in-time copy. Ex. 1016 ¶¶ 44, 118. Thus, each Snapshot copy would result in transferred data (i.e., database block) for different point-in-time copies being received and stored. Lastly, even if Patent Owner is correct that a given Snapshot copy uses only database blocks from the same point in time, we agree with Petitioner that this

does not mean that database blocks for different point-in-time copies are not received and stored.

We conclude that a preponderance of the evidence demonstrates that a person of ordinary skill would have found the limitation of loading the source database at multiple points in time, including receiving and storing database blocks for different point-in-time copies, obvious.

d) *“provisioning a virtual database”*

Patent Owner first argues that the proposed combination of references does not render obvious “provisioning a virtual database” even under Petitioner’s proposed construction, which, as discussed above, we adopt. PO Resp. 52–55. Specifically, Patent Owner argues that Sanders’s FlexClone volumes are not equivalent to virtual databases because they are not “a set of database files.” *Id.* at 52–53. Patent Owner asserts that inodes are not database files because “they are not capable of being mapped to physical addresses for stored database blocks,” but instead are “collections of inodes which may point to file system data blocks.” *Id.* at 54 (citing Ex. 2016 ¶ 149).

Patent Owner’s argument, however, is premised on the previously discussed definition of “database block.” As discussed above, we reject Patent Owner’s proposed construction.

Patent Owner next argues that the asserted references do not render obvious the “provisioning a virtual database” limitation under Patent Owner’s proposed construction. *Id.* at 55–56. This argument is not persuasive, however, because we do not adopt Patent Owner’s proposed construction, as discussed above.

We conclude that a preponderance of the evidence demonstrates that a person of ordinary skill would have found the claimed “provisioning a virtual

database” obvious over FlexClone volumes disclosed by Sanders and Edwards.

e) “*creating a set of files linked to the stored database blocks*”

Patent Owner argues that “Petitioner points to no disclosure in Edwards or Sanders (or any other document) in which a set of files are created and then linked to the stored database blocks of the source database.” PO Resp. 57. In particular, Patent Owner argues that Petitioner relies on Edwards’s disclosure of creating a container file, “[b]ut this ‘container file’ is not a newly created set of files.” *Id.* (citing Ex. 2016 ¶¶ 167–69). According to Patent Owner, the new “container file” of Edwards “is merely a copy of the ‘vol\_info’ block of the volume to be cloned with pointers to existing files,” and “[n]o new set of files is created.” *Id.* at 57–58 (citing Ex. 2016 ¶¶ 167–69).

Petitioner argues that “Edwards teaches that FlexClone creates a new ‘FlexVol volume,’” and “a FlexVol volume is a **file system** created *within* a file on an underlying file system.” Reply 21 (citing Ex. 1005, 11, 14). According to Petitioner, “cloning a volume containing database files creates a new FlexVol volume with a new file system comprising a new set of database files.” *Id.* at 22 (citing Pet. 36–39).

We find Petitioner’s argument persuasive. Edwards discloses that creating a clone volume, or FlexClone volume, “requires the writing of a fixed, small number of blocks,” and the process includes *creating the files* required for a new FlexVol volume by “seed[ing] 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. 1005, 14–15. Edwards discloses that a container file “contains all the blocks of the FlexVol volume”

and, in some respects, “serves as a virtual disk containing the FlexVol volume.” *Id.* at 11. Moreover, although a container file is initially sparsely populated, with “most of the logical offsets hav[ing] no underlying physical storage,” “WAFL allocates physical storage to the container file as the FlexVol volume writes data to new logical offsets.” *Id.* at 12. This disclosure suggests that a container file is a physical file rather than a mere copy of a block with pointers to actual files. And even if a container file is merely a copy of a block with pointers to actual files, as Patent Owner argues, the claim language does not distinguish between virtual and actual files. Rather, the challenged independent claims simply recite “a set of files linked to the stored database blocks.” As such, we find that creating container files creates a new set of files.

We conclude that a preponderance of the evidence demonstrates that a person of ordinary skill would have found the limitation of creating a set of files linked to the stored database blocks obvious.

*f) Conclusion*

In the Decision to Institute, we determined that Petitioner had shown a reasonable likelihood of prevailing on this proposed ground of unpatentability. Inst. Dec. 16–28. We have reviewed both parties’ arguments and supporting evidence, including the disclosure of all three references and the testimony of both Dr. Zadok and Dr. Shenoy. Despite the counter-arguments in Patent Owner’s Response, and the evidence cited therein, for the reasons discussed above, Petitioner has shown, by a preponderance of the evidence, that claims 1, 14, 17, and 24 are unpatentable under 35 U.S.C. § 103 as directed to subject matter that would have been obvious to a person of ordinary skill in the art in light of Sanders, Edwards, and Chapman.

5. *Dependent Claims*

a) *Claims 2 and 18*

Claim 2 depends from claim 1 and further recites that “the replicating of database blocks comprises transmitting the database blocks from the first storage system to the second storage system for storing in the second storage system.” Ex. 1001, 35:59–62. Similarly, claim 18 depends from claim 17 and further recites the same subject matter regarding the replicating of database blocks. *Id.* at 38:9–12. Petitioner contends that Chapman discloses the additionally recited limitations of claims 2 and 18. Pet. 38–39, 56 (citing Ex 1006, 10, 11; Ex. 1016 ¶¶ 144, 193, 194).

b) *Claim 3*

Claim 3 depends from claim 1 and further recites that “the replicating of database blocks comprises transmitting a subset of database blocks comprising database blocks that changed since a point-in-time from the first storage system to the second storage system for storing in the second storage system.” Ex. 1001, 35:63–67. Petitioner contends that Chapman discloses the additional limitations of claim 3. Pet. 39–40 (citing Ex 1006, 10, 11; Ex. 1016 ¶¶ 145, 146).

c) *Claim 4*

Claim 4 depends from claim 1 and further recites using the virtual database from the second storage system as a standby database when the source database is unavailable. Ex. 1001, 36:1–4. Petitioner asserts that Sanders discloses this limitation. Pet. 40–41 (citing Ex 1004, 5; Ex. 1016 ¶¶ 147, 148).

*d) Claims 5 and 19*

Claim 5 depends from claim 1 and further recites “transmitting database blocks stored in the second storage system to the first storage system” and “storing the database blocks received from the second storage system at the first storage system.” Ex. 1001, 36:5–9. Similarly, claim 19 depends from claim 17 and adds that the computer executable code further comprises instructions to carry out the same steps. *Id.* at 38:13–18. Petitioner asserts that Chapman discloses these limitations. Pet. 41–43, 56–57 (citing Ex 1006, 18; Ex. 1016 ¶¶ 150–52, 195, 196).

*e) Claim 6*

Claim 6 depends from claim 1 and further recites “receiving request to update database blocks associated with the VDB from the database server running on the system.” Ex. 1001, 36:10–13. Petitioner contends that Sanders discloses this limitation. Pet. 43–44 (citing Ex 1004, 3, 5, 29; Ex. 1016 ¶¶ 153, 155).

*f) Claim 8*

Claim 8 depends from claim 1 and further recites provisioning a second virtual database system in a manner similar to the claim 1 limitation of provisioning a virtual database. Ex. 1001, 36:25–32. Relying on the testimony of Dr. Zadok, Petitioner asserts that the combination of Sanders, Edwards, and Chapman meets the limitations of claim 8 for reasons similar to those asserted in connection with the claim 1 limitation of provisioning a virtual database. Pet. 44–46 (citing Ex. 1016 ¶¶ 156–61, 163–65).

*g) Claims 16 and 25*

Claim 16 depends from claim 14 and further recites linking a second source database, loading a plurality of point-in-time copies of the second source database, and performing backup of database blocks similarly to how such steps are carried out in claim 14 with respect to a first source database. Ex. 1001, 37:39–53. Similarly, claim 25 depends from claim 24 and adds that the computer executable code further comprises instructions to link a second source database, load a plurality of point-in-time copies of the second source database, and perform backup of database blocks. *Id.* at 40:8–25.

For the reasons relied on in connection with the parallel limitations of claims 14 and 24, Petitioner asserts that the combination of Sanders, Edwards, and Chapman renders claims 16 and 25 obvious. Pet. 51–54, 59, 60 (citing Ex. 1016 ¶¶ 182–86, 203, 204).

*h) Discussion*

As described above, Petitioner provides detailed explanations supported by the testimony of Dr. Zadok and specific citations to Sanders, Edwards, and Chapman indicating where in the references the limitations of dependent claims 2–6, 8, 16, 18, 19, and 25 are taught. Patent Owner does not separately argue these dependent claims, but relies on the same arguments as those advanced with respect to the independent claims, which arguments we have found unpersuasive for the reasons discussed above.

Upon considering all of the evidence of record, we determine that Petitioner has shown, by a preponderance of the evidence, that claims 2–6, 8, 16, 18, 19, and 25 are unpatentable under 35 U.S.C. § 103 as directed to subject matter that would have been obvious to a person of ordinary skill in the art in light of Sanders, Edwards, and Chapman.

#### IV. MOTION TO EXCLUDE

Patent Owner moves to exclude Petitioner's Exhibits 1019, 1027, 1031, 1032, 1034–1037, 1040–1044, 1046, 1047, 1049, 1053, 1054, 1056, 1058, 1059, 1063, 1065, 1066, and 1069. Paper 37 (“Mot. to Exclude”), 1. 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 movant, Patent Owner has the burden of showing that an Exhibit is not admissible. 37 C.F.R. § 42.20(c).

Patent Owner contends that Petitioner does not rely on Exhibits 1031, 1032, 1034, 1037, 1043, 1049, 1053, 1056, 1058, and 1063. Mot. To Exclude 1 n.1. Petitioner does not dispute this contention. See Paper 48. We, therefore, grant Patent Owner's motion as it pertains to Exhibits 1031, 1032, 1034, 1037, 1043, 1049, 1053, 1056, and 1063. We note, however, that Petitioner does rely on Exhibit 1058. See Reply 3. For reasons discussed below, we dismiss Patent Owner's motion as it pertains to Exhibit 1058. Further, Petitioner has moved, unopposed, to expunge Exhibit 1027 (see Paper 35), which motion we hereby grant.

Of the remaining objected-to Exhibits, our Final Written Decision discusses and relies on only Exhibits 1019, 1046, and 1047. We determine, for reasons set forth 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 the remaining additional Exhibits. Accordingly, Patent Owner's motion to exclude Exhibits 1035, 1036, 1040–1042, 1044, 1054, 1058, 1059, 1065, 1066, and 1069 is *dismissed* as moot.

According to Patent Owner, Exhibits 1019 and 1046 are inadmissible hearsay. Mot. to Exclude 9 (citing Fed. R. Evid. 801, 802). In particular, Patent Owner argues that Mr. Hernandez lacks personal knowledge regarding the documents about which he testifies, rendering his statements hearsay. *Id.* at 9–12 (referring to Exs. 1019, 1046); Paper 50, 4–5. We note that Mr. Hernandez’s personal knowledge comes from his review recognition of the documents as being published during his tenure as an employee of NetApp or during his tenure as an employee of Midwave, a certified distributor and reseller of NetApp. Ex. 1019 ¶¶ 6–18; Ex. 1046 ¶¶ 3, 6, 11. This is ample basis for him to testify with personal knowledge of the facts under FRE 602. We are not persuaded otherwise by Patent Owner’s characterization of Mr. Hernandez’s cross-examination testimony. *See* Mot. to Exclude 10–11 (citing Ex. 2032); Paper 39 (citing Ex. 2032)). Furthermore, Mr. Hernandez’s testimony consists of statements made by the Declarant while testifying in this proceeding—not “hearsay” (Fed. R. Evid. 801(c))—but sworn testimony that is subject to cross-examination. Indeed, as noted above, Patent Owner cross-examined Mr. Hernandez with respect to that testimony. We, therefore, deny Patent Owner’s motion to exclude Exhibits 1019 and 1046.

The only reason Patent Owner gives for excluding Exhibit 1047 is that it is “untimely under 37 C.F.R. § 42.23(b) and 37 C.F.R. § 42.123.” Mot. to Exclude 1 n.1. We find that Exhibit 1047, an article titled “Bigtable: A Distributed Storage System for Structured Data,” was properly submitted by Petitioner as evidence in rebuttal to Patent Owner’s claim construction argument for the term “database block” asserting that it must necessarily include metadata. In particular, Petitioner argues this article is relevant to Patent Owner’s assertion that “[i]t is well known to one skilled in the art that a

database block necessarily includes metadata.” See PO Resp. 19–23; Reply 11 (citing Ex. 1047).

#### V. ORDER

In consideration of the foregoing, it is hereby:

ORDERED that claims 1–6, 8, 14, 16–19, 24, and 25 of the ’361 patent have been shown to be *unpatentable*;

FURTHER ORDERED that Petitioner’s motion to expunge Exhibit 1027 is *granted*;

FURTHER ORDERED that Patent Owner’s motion to exclude evidence is *granted-in-part, dismissed-in-part* and *denied-in-part*;

FURTHER ORDERED that Exhibits 1031, 1032, 1034, 1037, 1043, 1049, 1053, 1056, and 1063 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.

IPR2015-00100  
Patent 8,566,361 B2

PETITIONER:

Robert Steinberg  
Jonathan Link  
S. Giri Pathmanaban  
LATHAM & WATKINS LLP  
bob.steinberg@lw.com  
jonathan.link@lw.com  
giri.pathmanaban@lw.com

PATENT OWNER:

J. David Hadden  
Saina S. Shamilov  
FENWICK & WEST LLP  
dhadden-ptab@fenwick.com  
sshamilov-ptab@fenwick.com