

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

ACTIFIO, INC.,
Petitioner,

v.

DELPHIX CORP.,
Patent Owner.

Case IPR2015-00025 & IPR2015-00026¹
Patent 8,161,077 B2

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

EASTHOM, *Administrative Patent Judge*.

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

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

I. INTRODUCTION

Petitioner, Actifio, Inc., filed a request for an *inter partes* review of claims 1–7, 9, 10, 18–23, 25, and 26 of U.S. Patent No. 8,161,077 B2 (Ex. 1001, the “’077 patent”) under 35 U.S.C. §§ 311–319. Paper 1 (“Petition” or “Pet.”). The Board instituted an *inter partes* review of these claims on asserted grounds of unpatentability for obviousness. Paper 11 (“Dec. on Inst.”). In related Case IPR2015-00026 involving the same parties, pursuant to Petitioner’s *inter partes* request, the Board also instituted *inter partes* reviews of claims 8, 11–13, 17, 24, 27–29, and 33 of the ’077 patent in IPR2015-00026 (“’026 IPR”) on asserted grounds of unpatentability for obviousness. See ’026 Paper 11 (“’026 Dec. on Inst.”). Based on a substantial overlap of arguments and evidence presented in the two cases, to administer the proceedings more efficiently, we exercise our authority under 35 U.S.C. § 315(d) to consolidate the two proceedings for purposes of issuing one final written decision.

The Board has jurisdiction under 35 U.S.C. § 6(c). This Final Written Decision is entered pursuant to 35 U.S.C. § 318(a) and 37 C.F.R. § 42.73. For the reasons discussed below, we determine Petitioner has shown by a preponderance of the evidence that claims 1–13, 17–29, and 33 of the ’077 patent are unpatentable.

After institution of trial, Patent Owner filed a Patent Owner Response (Paper 23, “PO Resp.”), to which Petitioner filed a Reply (Paper 32, “Pet. Reply”).² Subsequently, Patent Owner moved to exclude (Paper 46, “PO

² Unless otherwise indicated, we refer to public (including redacted) Papers and Exhibits filed in IPR2015-00025.

Mot. to Exclude”) certain Exhibits, Petitioner opposed (Paper 53, “Pet. Exclude Opp.”), and Patent Owner replied (Paper 56, “PO Exclude Reply”). Patent Owner also filed Motions for Observation on certain cross-examination testimony of Petitioner’s declarant, Dr. Erez Zadok (Paper 49, “Obs. Zadok”), to which Petitioner filed Responses (Paper 57 (“Obs. Resp. Zadok”) and Paper 55. Patent Owner also filed a Paper identifying allegedly untimely evidence and evidence and arguments beyond the scope of Petitioner’s Reply. Paper 58 (“Exclude Pet. Reply Evid.”). The parties filed similar Papers and Exhibits in the ’026 IPR (to be designated, as indicated above, with the “’026” prefix: for example, “’026 Paper” and “’026 Exhibit”).

A combined oral Hearing in this proceeding and related Cases IPR2015-00014, IPR2015-00016, IPR2015-00019, IPR2015-00034, IPR2015-00026, IPR2015-00050, IPR2015-00052, and IPR2015-00128 was held on January 14, 2016. A Transcript of the Hearing is included in the record as Paper 68 (“Tr.”).

A. Related Proceedings

The ’077 patent is the subject of *Delphix Corp. v. Actifio, Inc.*, No. 5:13-cv-04613-BLF (N.D. Cal.). In proceedings involving the same parties, the Board instituted *inter partes* reviews of U.S. Patent No. 8,150,808 B2 in Cases IPR2015-00014, IPR2015-00016, IPR2015-00019, and IPR2015-00034; U.S. Patent No. 8,548,944 B2 in Cases IPR2015-00050 and IPR2015-00052; U.S. Patent No. 8,566,361 B2 in Cases IPR2015-00100 and

IPR2015-00108; and U.S. Patent No. 8,468,174 B1 in Case IPR2015-00128.³

B. Asserted Grounds of Unpatentability

We instituted *inter partes* review on the following grounds of unpatentability under 35 U.S.C. § 103(a):

Claims Challenged	References
1, 3, 5–7, 9, 10, 18, 20–23, 25, and 26	Sanders ⁴ and Edwards ⁵
2, 4, and 19	Sanders, Edwards, and Patel ⁶
8, 11–13, 24, and 27–29	Sanders, Edwards, and Data ONTAP Guide ⁷
17 and 33	Sanders, Edwards, Data ONTAP Guide, and Sarma ⁸

³ Case IPR2015-00136 has been consolidated with IPR2015-00128.

⁴ Jawahar Lal & Roger Sanders, *DB2: Cloning a Database using NetApp FlexClone™ Technology*, Network Appliance Inc., IBM Toronto Lab, TR-3460 (Apr. 30, 2006) (“Sanders”) (Ex. 1013).

⁵ Edwards et al., *FlexVol: Flexible, Efficient File Volume Virtualization in WAFL*, PROCEEDINGS OF THE ANNUAL TECHNICAL USENIX CONFERENCE, 129–142 (June 22–27, 2008) (“Edwards”) (Ex. 1014).

⁶ Dipesh Patel and Generosa Litton, *Rapid Database Development and Deployment*, Network Appliance, Inc., WP-7014-0307 (Mar. 2007) (“Patel”) (Ex. 1016).

⁷ *Data ONTAP® 7.1 Data Protection, ONLINE BACKUP AND RECOVERY GUIDE*, Network Appliance, Inc. (Jan. 12, 2007) (“Data ONTAP Guide”) (’026 Ex. 1117).

⁸ Sarma et al., U.S. Patent No. 7,631,021 B2 (Dec. 8, 2009) (“Sarma”) (’026 Ex. 1118).

C. The '077 Patent

The '077 patent describes a system and method to create a virtual database, which involves obtaining multiple “point-in-time” (“PIT”) copies of the database to be virtualized. See Ex. 1001, Abstract. In one virtual database embodiment represented by Figure 2a, “production database system 110 . . . is the source of the database being virtualized” to create virtual database 220 using virtual database files stored in database storage system 100. *Id.* at 7:52–54. A reproduction of Figure 2a of the '077 patent follows:

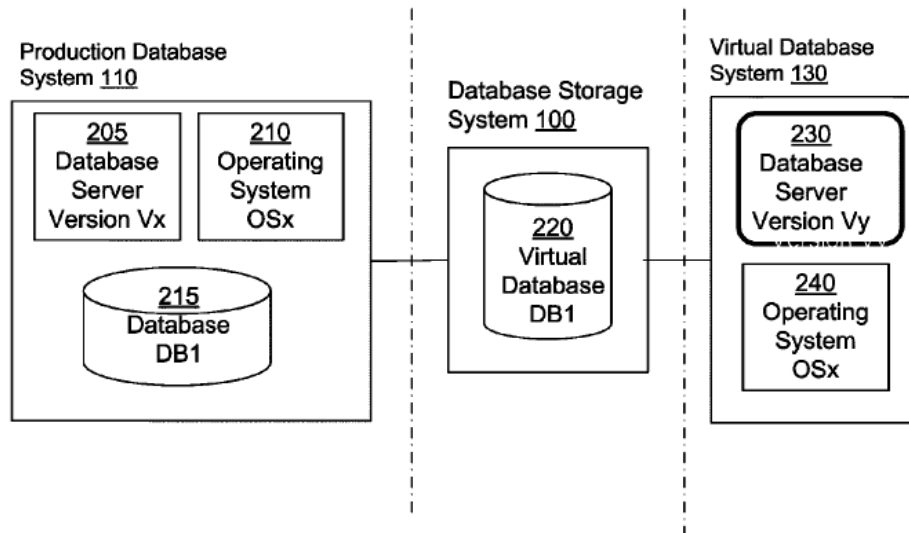


FIG. 2(a)

Figure 2a depicts production database system 110, virtual database DB1 220, and virtual database system 130, which accesses virtual database 220.

To virtualize a production database, the system of the '077 patent makes a first PIT copy of the production database and stores an entire set of database blocks representing the production database at that time in database storage system 100. See Ex. 1001, 19:23–24, Figs. 10–13. Subsequent PIT copies involve incremental changes and copy “only the blocks that changed

since the last PIT copy and may copy much less data compared to the first PIT copy.” *Id.* at 19:27–29.

The system creates a virtual database (VDB) by creating VDB file structures comprising VDB blocks that point to different PIT database blocks. *See id.* at 19:15–50. For example, the system creates VDB file structures 1050 that point to different PIT database blocks F11, G11, F22, etc. *See id.* at 19:15–50; Fig. 10. Each time database storage system 100 receives an updated PIT copy reflecting database block changes in the production database, the system updates the appropriate VDB blocks in a VDB file to be “implemented as pointers to the actual database block that stores the [updated] data.” *See id.* at 19:34–37.

A reproduction of Figure 10 from the ’707 patent follows:

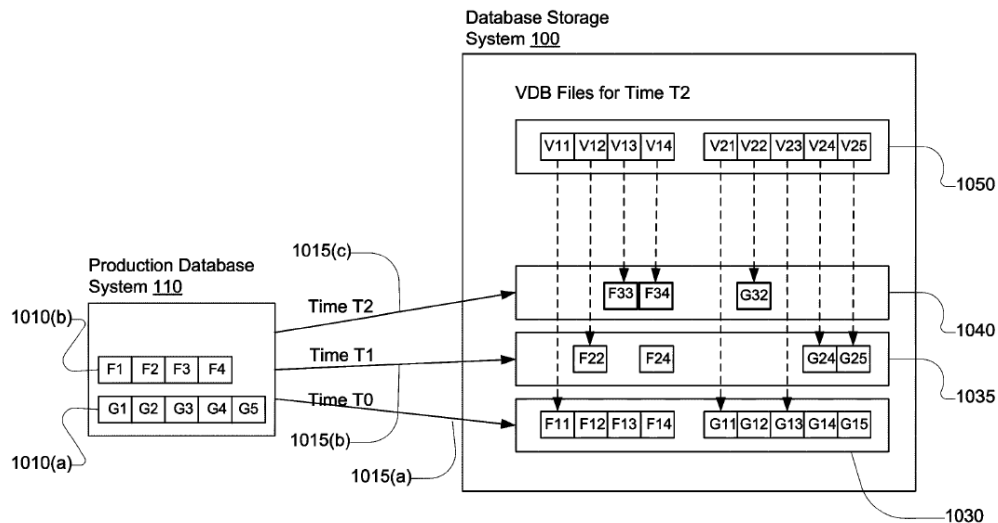


FIG. 10

Figure 10 shows “VDB Files for Time T2” in database storage system 100. Figure 10 further shows that “VDB file structures 1050” includes blocks V11, V12, V13, and V14 which point to database blocks F11 . . . F34 that

represent different PIT (i.e., at times T0, T1, and T2) copies of production database blocks F1, F2, F3, and F4 at production database system 110. *See* Ex. 1001, 19:33–34. Initially, all the production database blocks are copied to create “[t]he first PIT copy 1030 made at time T0,” as represented by database blocks F11, F12, F13, F14, G11, G12 . . . G15 in database storage system 100. *Id.* at 19:23–24. Later, when database storage system 100 receives the PIT copy at time T2 and updates the VDB blocks, block V13 points to the updated data at block F33, which represents a change existing at T2 to the data in block F3 in the production database (*see id.* at 19:41–43), whereas VDB block V11 still points to the data in block F11 “since the [production database] block F1 was never updated during copies made at time T1 and T2” (*id.* at 19:38–39).

D. Illustrative Claim

Of the challenged claims, clams 2–7, 9, and 10 depend directly or indirectly from independent claim 1. Claims 19–23, 25, and 26 depend directly or indirectly from independent claim 18, which closely tracks independent claim 1, with claim 18 reciting a “computer program product” instead of a method. *See* Ex. 1001, 46–47 (Certificate of Correction).

Illustrative claim 1 follows:

E. Challenged Claim 1

1. A method for test and development of databases and database applications using a virtual database system, the method comprising:

linking a source database, wherein linking the source database comprises receiving information identifying the source database;

loading a plurality of point-in-time copies of the source database, wherein the loading comprises:

receiving database blocks for the point-in-time copies of the source database, and

storing the database blocks on a storage system;
and

provisioning a development virtual database to a development system and a test virtual database to a test system, wherein provisioning each of the virtual databases to a system comprises:

creating a set of files linked to the stored database blocks on the storage system, and

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

II. ANALYSIS

A. Patent Owner's Motion to Exclude Evidence

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

1. The '025 IPR

Patent Owner moves to exclude Petitioner's Exhibits 1029, 1030, 1032, 1033, 1035–1146, 1048, 1049, 1054, and 1062. PO Mot. to Exclude 1. As Patent Owner notes, however, Petitioner does not rely on Exhibits

1030, 1032, 1033, 1035–1137, 1040, 1041, and 1045. *Id.* at 1 n.1. Of the other objected-to Exhibits, except for Exhibits 1048 and 1049, we do not, and need not, consider such evidence. We determine, for reasons set forth below, 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 that rely on those additional Exhibits. Accordingly, Patent Owner’s Motion to Exclude Exhibits 1029, 1030, 1032, 1033, 1035–1146, 1054, and 1062 is *dismissed* as moot.

Exhibit 1048 is the Supplemental Declaration of Louis Hernandez. Patent Owner argues that the Exhibit is “inadmissible hearsay.” PO Mot. to Exclude 8–9. The Supplemental Declaration, however, 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, Patent Owner cross-examined Mr. Hernandez with respect to that testimony.⁹

⁹ Patent Owner’s allegation of “double hearsay” is not persuasive. PO Mot. to Exclude 8; PO Exclude Reply 4–5. Patent Owner does not argue that it objected to any NetApp documents that Mr. Hernandez relied upon as hearsay. *See* Pet. Exclude Opp. 13; ’026 Pet. Exclude Opp. 13 (Paper 12 (contending no hearsay objection by Patent Owner to the NetApp documents at issue: i.e., Sanders, Patel, and Data ONTAP Guide)); PO Exclude Reply 4–5 (replying to Petitioner’s contention but not disputing a lack of an objection). In addition, as discussed herein and further below, Mr. Hernandez relies on document dates, other indicia, and his knowledge of NetApp’s standard practices about dated NetApp documents, not merely dates on documents. *See* Pet. Reply 4–5 (citing Ex. 1020 ¶ 17; 1048 ¶ 7); Pet. Exclude Opp. 7–13 (citing Ex. 1020 ¶ 6; Ex. 1048 ¶¶ 7, 10; Ex. 2028, 42:24–37, 47:5–11, 94:12–95:3, 101:17–102:9 (Hernandez Deposition testimony)).

Patent Owner also argues that Mr. Hernandez lacks personal knowledge to testify. Mr. Hernandez’s personal knowledge of practices about NetApp document publications comes from his review and recognition of documents published before and during his tenure as an employee of NetApp. *See* Ex. 1020 ¶¶ 1–6; Ex. 1048 ¶¶ 3–11. On this record, ample basis exists 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 direct and cross-examination testimony. *See* PO Mot. to Exclude 9–11.

Patent Owner moves to exclude Exhibit 1049 as untimely. Mot. to Exclude 1, n.1 (citing 37 C.F.R §§ 42.23 (b), 42.123). Contrary to this argument, Petitioner cites this document as rebuttal evidence in response to Patent Owner’s claim construction (i.e., to show that a well-known database does not use metadata in database blocks and uses metadata external to the block). *See* Pet. Reply 9–10 (arguing that Patent Owner’s preliminary database block construction at a district court did not include metadata, and that the alleged metadata requirement only occurred after Petitioner filed “the last of its IPR petition[s],” citing Ex. 1022, 6 (database block construction after e-mail chain of Oct. 24, 2014)); PO Resp. 24, nn.6–7 (citing Ex. 2025, 4 (joint district court claim construction, Oct. 27, 2014)).

Patent Owner’s Motion to Exclude Exhibits 1048 and 1049 is *denied*.

2. The ’026 IPR

Patent Owner moves to exclude Petitioner’s ’026 Exhibits 1117, 1134–1136, 1138–1148, 1150, 1151, 1153, 1156, and 1164. ’026 Paper 45 (“Motion to Exclude”), 1. As Patent Owner notes, however, Petitioner does

not rely on '026 Exhibits 1134–1136, 1138–1142, and 1147. *Id.* at 1 n.1. Of the other objected-to Exhibits, except for '026 Exhibits 1117 and 1150, we do not, and need not, consider such evidence. We determine, for reasons set forth below, 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 that rely on those additional Exhibits. Accordingly, Patent Owner's motion to exclude '026 Exhibits 1134–1136, 1138–1148, 1150, 1151, 1153, and 1156 is *dismissed* as moot.

Patent Owner argues that '026 Exhibit 1150 (Supplemental Declaration of Louis Hernandez) is “inadmissible hearsay,” and that Mr. Hernandez lacks personal knowledge to testify regarding publication. *See* '026 Paper 45, 1. Based on the determination above, '026 Exhibit 1150, which is a copy of Exhibit 1048 discussed above, is admissible.

According to Patent Owner, '026 “Exhibit 1117 is allegedly a user manual for a NetApp product which Petitioner relies upon for its prima facie case of obviousness.” '026 Paper 45, 1–2. Patent Owner moves to exclude the Exhibit (Data ONTAP Guide) as not being authenticated pursuant to Federal Fed. R. Evid 901. *Id.*

Petitioner provides reasons why it contends Data ONTAP Guide is self-authenticating under Fed. R. Evid 902 (7). '026 Paper 51, 13–14. A document may be authenticated by “the appearance, contents, substance, internal patterns, or other distinctive characteristics of the item, taken together with all the circumstances.” Fed. R. Evid. 901(b)(4). In addition to relying on NetApp trademark symbols, copyright notices, and other indicia (*id.* at 13), Petitioner submits the following:

The cover page includes NetApp's company address, telephone number, website, and email address for providing comments to NetApp about the documents. Indeed, the website and email address on Ex. 1117 are hyperlinked, and clicking on them directly links a user to NetApp's website or opens an email addressed to NetApp respectively. Further, the Data ONTAP Guide contents include repeated references to NetApp and various NetApp technologies. These distinctive characteristics are more than sufficient for authentication.

Id. at 14. Moreover, Petitioner, relying on the testimony of Mr. Hernandez, points out evidence and testimony that is sufficient to authenticate the document pursuant to Rule 901(b)(1). See '026 Paper 45, 13–14. Contrary to Patent Owner's argument, Mr. Hernandez has sufficient knowledge to authenticate Data ONTAP Guide. See '026 Paper 45, 13–14; '026 Paper 51, 2; '026 Ex. 1122 ¶¶ 1–6, 10 (Data ONTAP Guide); '026 Ex. 1150.

We, therefore, are not persuaded that Data ONTAP Guide is not authenticated at least under Federal Rules of Evidence 901(a), (b)(1), (b) (4), and 902 (7). On this record, we accept '026 Exhibit 1117 for what it purports and is alleged to be: A “Data Protection Online Backup and Recovery Guide” for a NetApp product, published by NetApp, for the purpose of providing detailed instructions to its customers and potential customers. See '026 Paper 45, 13; '026 Ex. 1117, 1 (emphasis omitted).

Patent Owner's Motion to Exclude '026 Exhibits 1117 and 1150 is *denied*.

B. Printed Publication—Sanders, Patel, and Data ONTAP Guide

Patent Owner contends that of Sanders, Patel, and Data ONTAP Guide are not a prior art printed publications in accordance with 35 U.S.C. §§ 102 and 311(b). PO Resp. 1; '026 PO Resp. 1. Underlying facts inform

the 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).

Initially, Patent Owner’s contention that Petitioner cannot rely upon evidence not submitted with the Petition to show that the documents are not prior art is not supported. PO Resp. 3–5; Paper 58; ’026 PO Resp. 3–5. In Patent Owner’s view, Petitioner must make out a *prima facie* case of unpatentability in its Petition, which includes the substantive element of the documents being publicly accessible and prior art. PO Resp. 4; Paper 58, 1; ’026 PO Resp. 4. That position, correct as stated, fails to account for the difference between the threshold for instituting a trial, 35 U.S.C. § 314(a), and proving unpatentability of a claim in trial, 35 U.S.C. § 316(e). As noted by our reviewing court, “there is a significant difference between a petitioner’s burden to establish a ‘reasonable likelihood of success’ at institution, and actually proving invalidity by a preponderance of the evidence at trial.” *TriVascular, Inc. v. Samuels*, 812 F.3d 1056, 1068 (Fed. Cir. 2016) (quoting 35 U.S.C. § 314(a) and comparing § 316(e)).

Based on the information presented in the Petitions and Patent Owner's Preliminary Responses, we determined there was a reasonable likelihood that Petitioner would prevail in its challenges that included the three publications. 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 Responses. Patent Owner, in fact, stated that "the relevant NetApp features were disclosed to the Patent Office during the prosecution of the challenged claims." Paper 10, 2, 54 (characterizing references as "redundant references describing those same NetApp features" disclosed during prosecution and stating that Patent Owner "disclosed to the Patent Office every NetApp feature that Petitioner now cites in the Petition"); '026 Paper 10, 2, 54 (same). We do not 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 the three challenged prior art disclosures, 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 two specific Declarations in each Reply. PO Resp. 2–3 (citing Ex. 1020; Ex. 1028); '026

PO Resp. 23 (citing Ex. 11[2]2; Ex. 1130).¹⁰ Exhibit 1028 and '026 Ex. 1130 are no longer relevant.¹¹ In the relevant set of Declarations, Louis Hernandez provides the same “Hernandez Declaration” in each case: i.e., Ex. 1020 and '026 IPR, Ex. 1122.

Patent Owner submits that Petitioner provided Patent Owner with the Hernandez Declaration in response to Patent Owner’s objections to evidence (although Patent Owner does not tell us its basis for the objections). PO Resp. 2; '026 PO Resp. 2. Board rules authorize serving supplemental evidence in response to an objection. 37 C.F.R. § 42.64(b)(2). Accordingly, Patent Owner lacks a basis to complain that evidence has been produced and served in response to its objections. *See* Pet. Reply iv (stating Ex. 1020 was served on May 12, 2015). Petitioner also relies, properly, on the supplemental evidence in its Replies (i.e., as evidence in reply to Patent Owner’s arguments in its Response that the three references are not printed publications).

Turning to the substance of the Hernandez Declaration, 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. 1020 ¶¶ 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

¹⁰ Patent Owner cites Exhibit 1112 instead of Exhibit 1122—an apparent typographical error. *See* '026 IPR, PO Resp. 3.

¹¹ These Exhibits (Declarations of Joseph Ortiz) are expunged at Petitioner’s request. Paper 43; '026 Paper 42.

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 all three documents, and that there were published during his tenure at NetApp. *Id.* ¶¶ 7, 10 (Data ONTAP Guide), 11 (Sanders), 12 (Patel).

Patent Owner argues Mr. Hernandez does not declare that Sanders, Patel, and Data ONTAP Guide, and were “publicly accessible.” PO Resp. 2; ’026 IPR, PO Resp. 2. Patent Owner submits the following: 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. PO Resp. 2–3; ’026 PO Resp. 2–3.

Petitioner replies with a Supplemental Declaration from Mr. Hernandez. Pet. Reply 6–7 (citing Ex. 1048); ’026 Pet. Reply 6–7 (citing ’026 Ex. 1150).¹² Mr. Hernandez testifies that he uses the term “publish” or “published” as referring to documents being publicly distributed to customers, potential customers, business partners, and alliances as of the month and year that appeared on the face of the documents, non-

¹² As indicated above, Exhibit 1048 and ’026 Ex. 1150 appear to be identical in testimony.

confidentially. Ex. 1048 ¶ 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 reflects 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. Pet. Reply 6; Ex. 1058, 3; ’026 Pet. Reply 6; ’026 Ex. 1163, 3.¹³

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 the Hearing Transcript (“Tr.”), Patent Owner’s Motion for Observation on Cross-Examination Testimony of Mr. Hernandez (Paper 47; ’026 Paper 46) and Petitioner’s Response to Owner’s Motion for Observation on Cross-Examination Testimony of Mr. Hernandez (Paper 57; ’026 Paper 52), insofar as they relate to public accessibility of Sanders, Patel, and Data

¹³ We find that Exhibit 1058 (and corresponding ’026 Ex. 1163), 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.

ONTAP Guide.¹⁴ 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 the three NetApp documents to be credible. Sanders purports to be a technical report ("TR-3460 . . . April 30, 2006") showing how to use "NetApp FlexClone™ Technology" so that "you can create an exact copy of a DB2 database," using a NetApp solution "that helps database and system administrators" "to create an instant clone." Ex. 1013, 1, 3 (emphasis omitted). Patel purports to be a "March 2007" "NetApp White Paper" ("WP-7014-0307") and describes similar NetApp product "solutions" for "midsize businesses to global corporations" that [d]atabase administrators and applications developers can use" as "simple but powerful tools to quickly and easily create local or remote database clones in seconds, using little additional storage." Ex. 1016, 1, 3. Data ONTAP Guide purports to be a NetApp "Data Protection Online Backup and Recovery Guide," with "Part number 210-02020_A0 Updated for Data ONTAP 7.1.2 on 12 January 2007," and includes Data ONTAP 7.1 nomenclature. '026 Ex. 1117, 1 (emphasis omitted). Dr. Zadok points out that Sanders expressly refers to Data ONTAP Guide. See Ex. 1013, 50; '026 Ex. 1109 ¶ 109. Sanders lists a website for it. Ex. 1013, 50.

As an earlier panel of the Board has found, a dated technical document, having no indication of being a mere draft or an internal paper, is

¹⁴ During the Hearing, Patent Owner asked for, and we granted, additional time to consider its oral Hearing arguments regarding alleged new issues (regarding publication) raised in Petitioner's Reply in lieu of filing a Sur-Reply. See Tr. 211:16–212:18; 224:13–21; 237:1–25.

“a type of document whose very purpose is public disclosure.” *Veeam Sw. Corp. v. Symantec Corp.*, Case IPR2014-00089, slip op. at 14 (PTAB Apr. 25, 2014) (Paper 9).

Further regarding Sanders, according to Petitioner, in response to Patent Owner’s objections,

Petitioner timely provided a declaration from the office manager of the Internet Archive (Wayback Machine). Ex. 1023, Affidavit of C. Butler at ¶¶ 1–6, pp. 0231-292. Based on the recorded dates, Sanders was publicly archived by Wayback from NetApp’s website by November 22, 2006. *See* Ex. *id.* at p. 0231 (showing date of “20061122”); *id.* 1023 ¶ 5 (explaining [yyyy][mm][dd] format of dates in archived URLs). An affidavit from the Internet Archive is sufficient to show public availability as of recorded dates listed in the URLs. *Par Pharma v. Jazz Pharma*, IPR2015-00546, Paper 25 at 25.

Pet. Reply 4. Exhibit 1023, Affidavit of Cristopher Butler, supports Petitioner regarding Wayback’s archival of Sanders. Ex. 1023 ¶¶ 1–6, pp. 231–92, 223 (listing “TR-3460 DB2: Cloning a Database using NetApp FlexClone Technology (PDF)” (emphasis deleted)).

“A given reference is ‘publicly accessible’ upon a satisfactory showing that such document has been disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art exercising reasonable diligence, can locate it.” *SRI Int’l, Inc. v. Internet Sec. Sys., Inc.* 511 F.3d 1186, 1194 (Fed. Cir. 2008) (quoting *Bruckelmyer v. Ground Heaters, Inc.*, 445 F.3d 1374, 1378 (Fed. Cir. 2006)). In view of the foregoing considerations, we find that Petitioner has established, by a preponderance of the evidence, that Sanders (April 30, 2006), Patel (March, 2007) and Data ONTAP Guide (Jan. 12, 2007) were

available to the interested public at least more than one year before October 21, 2009, the earliest possible priority date of the '077 patent. Therefore, on this record, Sanders, Patel, and Data ONTAP Guide are printed publications under 35 U.S.C. §§ 102 and 311(b).

C. Claim Construction

In an *inter partes* review, claim terms in an unexpired patent are given their broadest reasonable construction in light of the specification of the patent in which they appear. 37 C.F.R. § 42.100(b); *see also In re Cuozzo Speed Techs., LLC*, 793 F.3d 1268, 1277–79 (Fed. Cir. 2015) (“Congress implicitly approved the broadest reasonable interpretation standard in enacting the AIA,” and “the standard was properly adopted by PTO regulation.”), *cert. granted sub nom. Cuozzo Speed Techs., LLC v. Lee*, 136 S. Ct. 890 (mem.) (2016). In general, claim terms presumptively carry their ordinary and customary meaning in view of the specification, as would be understood by one of ordinary skill in the art at the time of the invention. *In re Translogic Tech. Inc.*, 504 F.3d 1249, 1257 (Fed. Cir. 2007); *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312–13, 1315 (Fed. Cir. 2005) (*en banc*). A patentee may rebut that presumption by providing a definition for the term in the specification with reasonable clarity, deliberateness, and precision. *In re Paulsen*, 30 F.3d 1475, 1480 (Fed. Cir. 1994). A claim generally does not incorporate features of a particular embodiment appearing in the written description if the claim recites broader language or otherwise does not recite those features. *SuperGuide Corp. v. DirecTV Enterprises, Inc.*, 358 F.3d 870, 875 (Fed. Cir. 2004); *In re Van Geuns*, 988 F.2d 1181, 1184 (Fed. Cir. 1993); *Phillips*, 415 F.3d at 1323 (“[A]lthough the specification often

describes very specific embodiments of the invention, we have repeatedly warned against confining the claims to those embodiments.”).

During trial, the parties disputed the claim construction of the terms “database block” and “virtual database,” as addressed below. No other claim terms require express construction to resolve the issues raised in this *inter partes* review. See *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999) (only disputed claim terms need to be construed, and only to the extent necessary to resolve a material issue in dispute).

1. Database Block

i) Whether a Database Block Requires Metadata

The main claim construction dispute between the parties with respect to the term “database block” centers on whether a database block must necessarily include metadata. Patent Owner asserts the term “database block” should be interpreted to require metadata, i.e., as “a unit of data used by a database which comprises a specific number of bytes stored in the storage, a portion of which stores metadata associated with the unit of data.” PO Resp. 17. Petitioner argues the correct interpretation of the term is “a unit of data used by a database.” Pet. 10 (emphasis omitted). For the reasons discussed below, we conclude the disputed term is not limited as Patent Owner contends.

Claim construction analysis begins by considering the language of the claims themselves. *Phillips*, 415 F.3d at 1314. None of the challenged claims recite the term “metadata.” Only relevant claims 32 and 33 of U.S. Patent No. 8,150,808 B2 (the “’808 patent”) in related Cases IPR2015-

00014, IPR2015-00016, IPR2015-00019, and IPR2015-00034, recite “metadata.” See Ex. 2001, 44–45. Claims 32 and 33 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 in the case at hand, or any other challenged claims to require metadata, they could have done so by explicitly modifying the disputed term with “metadata,” but did not.

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

Patent Owner asserts that a definition of “database block” appearing in the “related ’808 [patent] specification” applies to the case at hand. PO Resp. 17–18. Patent Owner asserts that the following passage in the Summary section of the ’808 patent specification defines the term “database block”:

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

Id. at 17–18 (quoting Ex. 2001, col. 2:7–12).

Not employing the passage in the '077 patent itself implies that there is no clear intent to limit the claims in the '077 patent. In any event, assuming that the passage somehow has written description support in, and applies to, the '077 patent, the first sentence in the cited passage above explicitly defines the term, by stating “[a] database block is a unit of data used by a database.” Ex. 2001, 2:9–11.¹⁵ Although the second next sentence states that a database block “stores metadata,” that sentence by itself is insufficient to limit the disputed term by requiring the unrecited “metadata” feature, because it does not state unambiguously that all “database blocks” must include metadata. As discussed below, other portions of the Specification indicate that some database blocks do not include metadata. *See Liebel-Flarsheim*, 358 F.3d at 906 (construing a claim term broadly because “[n]o statement in the written description [] constitute[d] a limitation on the scope of the invention”) (quoting *Brookhill-Wilk 1, LLC. v. Intuitive Surgical, Inc.*, 334 F.3d 1294, 1301 (Fed. Cir. 2003)); *see also id.* at 908 (passages in the Summary of the Invention section of a patent do 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”).

¹⁵ 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 that this addition is not part of an explicit definition, rather, it represents embodiments within the defined term. Furthermore, Patent Owner fails to explain how to interpret that phrase or how it presents a material issue related to the prior art.

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

Patent Owner asserts, citing certain portions of the Specification and the Declaration of Prashant Shenoy, Ph.D. (Ex. 2017, “Shenoy Declaration”), that if a database block does not include metadata, the system disclosed in the ’077 patent would not work as described. PO Resp. 18–19. For example, Patent Owner argues that a database block must include

metadata because the disclosed system analyzes the metadata of each block to store only the incremental changes made to the production database. *Id.* at 19 (citing Ex. 1001, 14:23–24, 32–35). Patent Owner contends that storing such incremental changes “is one of the main functions” of the claimed system essential to achieving “a main purpose” of the invention—“to efficiently provide virtual database . . . without proliferating redundant copies of database data.” *Id.* at 19 (citing Ex. 1001, 3:5–7, 7:22–26; 7:32–35; 8:41–49 Ex. 2017 ¶¶ 63–67, 76, 79, 80). 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 19–20 (citing Ex. 1001, 15:15–19; Ex. 2017 ¶¶ 79, 84).

Patent Owner’s arguments are unpersuasive because the described system will work without requiring metadata in the claims, and any 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.”).

Moreover, the ’077 patent Specification explicitly makes a distinction between database blocks and other stored information, including metadata: “The database storage 100 system retrieves information The *information* retrieved *includes databask blocks* comprising data stored in the database, transaction log information, *metadata* information related to the database, information related to users of the database and the like.” Ex.

1001, 9:29–35 (emphasis added). Furthermore, as noted by Petitioner, the ’707 patent (and the ’808 patent) discloses an alternative embodiment where the transfer of production database data is achieved by “using a *file sharing system* similar to the file sharing system 120,” such as a network file system (NFS). Pet. Reply 8 (quoting Ex. 1001, 8:52–56; citing Ex. 1031, 67:18–68:7, 192:13–19; Ex. 1063 ¶¶ 9–11). As discussed below, the written description does not show that the file sharing embodiment requires metadata.

In the streaming embodiment relied upon by Patent Owner, as represented by Figures 4 and 5 and generally described in column 13, line 7 to column 16, line 8 of the ’707 patent, the production database system, upon receiving a request for data from the PIT copy manager of the database storage system (Ex. 1001, 13:19–31), packages the production database data “into a format that can be processed by the PIT copy manager” (*id.* at 13:50–53) and builds the appropriately formatted data into a data stream that is sent to the PIT copy manager. *Id.* at 13:55–61; *see id.* at 4:23–30 (describing Figs. 4 and 5 as relating to “an embodiment of the invention” “for processing a stream of data”). Upon receiving the data stream, the PIT copy manager processes the data stream to identify database blocks contained in it. *Id.* at 14:23–31. In this data streaming embodiment, “[e]ach database block includes metadata,” which is used to “identify database blocks . . . in the stream of data.” *Id.* at 14:22–25. When saving a retrieved database block into a transferred or copied database file on the database storage system, the PIT copy manager “analyzes the database block metadata to map

. . . the database block to [the] database file and a location within the file.”
Id. at 15:17–19.

Notwithstanding this embodiment, we find nothing that limits the invention as Patent Owner contends. For example, it may be necessary to include metadata in each database block transmitted on a data stream in order to identify and unpack database blocks from a continuous stream of data that has no apparent structure or boundaries. Nevertheless, the streaming embodiment does not show that a 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., id.* at 13:62 (describing “other embodiments”); 14:1–5 (“In these embodiments, 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);¹⁶ 7:3–10 (“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).

In the file sharing system, the database file on the production system can be accessed and copied directly by “mount[ing] the production DB data

¹⁶ The phrase “metadata *associated* with database blocks” implies, like the passage quoted above, that any metadata, may be, but need not be, *in* the database blocks. *See* Ex. 1001, 9:29–35 (describing database blocks as separate information from metadata).

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 8:49–56. Furthermore, contrary to Patent Owner’s argument, analyzing the metadata of each database block in the data stream is not necessary to achieve incremental updates in each embodiment. In the data stream embodiment, the data stream may include unnecessary database blocks, such as the blocks that did not change since the last PIT copy was transmitted, which are eliminated after the data stream is received at the database storage system by analyzing metadata for each database block. *Id.* at 13:32–42; 14:32–53. In an alternative embodiment not addressed by Patent Owner, the unchanged blocks are eliminated at the production system and never sent to the database storage system. *Id.* at 14:53–59 (“In [sic] *other embodiment*, some or all of the unnecessary blocks may be *eliminated while the data stream is built* by the production system library In this embodiment, the data stream . . . is reduced in size resulting in efficient communication between [the production system and the database storage system].”) (emphases added).

Patent Owner does not explain why metadata must be included in each database block to achieve the incremental update function in this embodiment. Therefore, the Specification does not indicate that copying database files by streaming data constitutes the essence of the claimed invention rather than a preferred embodiment. Such a preferred embodiment 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).

In addition, the '077 patent Specification generally states “*structures and functionality presented as a single component may be implemented as separate components.*” Ex. 1001, 32:29–32 (emphasis added); Pet. Reply 12. As Petitioner points out, the '077 patent specifically describes “*structures Fi and Gi [that] represent database blocks stored in the files 1010 respectively.*” Pet. Reply 12 (quoting Ex. 1001, 19:19–20) (emphasis by Petitioner). In the context of a database block, these disclosures bolster the disclosure cited above (*see, e.g., supra* note 16, Ex. 1001, 15:33–35 (discussed below)) and show that metadata may be separate from or combined with a database block, as Petitioner argues. *See* Pet. Reply 12 (citing Ex. 1063 ¶ 31).

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 and 18 recite receiving PIT copies of a source database and storing database blocks associated with the received PIT copies on a storage system. But the claims do not say anything about a particular method of transferring PIT copies, whether by streaming or by file sharing. Hence, the claims cannot be limited to either embodiment, and, therefore, the streaming embodiment cited by Patent Owner does not limit the claims.

In addition, consideration of differences among the claims of the '808 patent (upon which Patent Owner relies as noted above), supports the conclusion that Patent Owner’s cited passages are not limiting. Claim 29, which depends from claim 1 of the '808 patent, recites “receiving point-in-

time copies” by “receiving data streams” which comprise “data from database blocks.” Claims 32, 33, and 34 each depend from claim 29 and additionally recite “identify[ing] database blocks” in the data streams, “analyzing the metadata of database blocks to determine the length of the database blocks” (claim 32), “analyzing the metadata of database blocks to determine whether the database block needs to be stored” (claim 33), and “determining not to store the database blocks that . . . did not change since a previous retrieval of point-in-time copy” (claim 34). *See* Ex. 2001, 45–46. Hence, the subject matter specifically claimed in these dependent claims corresponds to the written description in the Specification relating to the streaming embodiment discussed above. *See* Ex. 2017 ¶¶ 79–80; PO Resp. 18–19 (citing Ex. 2017 ¶¶ 76, 79–80).

Claim 1 of the ’077 patent, which is similar to claim 1 of the ’088 patent, is presumed to be broader and not limited by the additional limitations relating to streaming and metadata recited in these dependent claims of the related ’088 patent. “[I]n a situation where dependent claims have no meaningful difference other than an added limitation, the independent claim is not restricted by the added limitation in the dependent claim.” *Trustees of Columbia Univ. in City of New York v. Symantec Corp.*, No. 2015-1146, 2016 WL 386068, at *10 (Fed. Cir. Feb. 2, 2016) (citing *Phillips*, 415 F.3d at 1314–15; *Acumed LLC v. Stryker Corp.*, 483 F.3d 800, 806 (Fed. Cir. 2007)).

In addition to the citations described above that show that database blocks do not require metadata, the ’077 patent describes storing database blocks as follows: “The file in which the database block is saved comprises

a file header including metadata associated with the file and a sequence of database blocks.” Ex. 1001, 15:33–35. This passage shows that the metadata can be stored in the file header separately from the series of database blocks stored in the body of the file. In sum, the Specification does not clearly and consistently describes the claimed invention as requiring metadata in database blocks.

Therefore, in view of the entire disclosure of the ’707 patent and the plain language of the claims, we find, notwithstanding the statement in the summary section of the ’808 patent relied upon by Patent Owner, “[a] portion of the database block stores metadata associated with the database block” (Ex. 2001, 2:10–12), that the record does not justify limiting the term “database block” by reading in the “metadata” limitation not found in the claims. *See Liebel-Flarsheim*, 358 F.3d at 908; *MEMS Tech. Berhad*, 447 F. App’x at 151.

Patent Owner also argues additional evidence supports a limiting construction. For example, citing the testimony of Dr. Shenoy and the testimony of Petitioner’s expert, Dr. Zadok (Ex. 1019, “Zadok Decl.”), Patent Owner asserts that requiring metadata in database blocks “comports with how the term is used by every major database system provider.” PO Resp. 18 (citing Ex. 2017 ¶¶ 41–46, 77). Patent Owner also argues that databases mentioned in the ’077 patent, such as Microsoft SQL and IBM DB2 (Ex. 1001, 5:55–67), require metadata in database blocks. *See* PO Resp. 23 (citing Ex. 2018 ¶ 158, n.11), n.5 (citing Ex. 2019, 58–59, n.7). In the paragraphs cited by Patent Owner, Dr. Shenoy discusses various documents describing some of the database systems listed in the ’077 patent,

including Oracle, Sybase, Microsoft SQL Server, and IBM DB2, and testifies that these database systems all require metadata in database blocks.¹⁷ Ex. 2017 ¶¶ 43–46, n.11.

The extrinsic evidence cited by Patent Owner is “less significant than the intrinsic record in determining the legally operative meaning of claim language.” *Phillips*, 415 F.3d at 1317 (citations omitted) (internal quotation marks omitted). Even if the external documents discussed by Dr. Shenoy describe embodiments of the ’707 patent, the evidence would be insufficient to limit the term “database block,” because even if the ’077 does not describe database blocks that do not require metadata (it does), reading in the “metadata” limitation not found in the claims from preferred embodiments is improper. *See Cadence Pharm. Inc. v. Exela PharmSci Inc.*, 780 F.3d 1364, 1369 (Fed. Cir. 2015) (“[E]ven if *all of the embodiments* discussed in the patent included a specific limitation, it would not be proper to import from the patent’s written description limitations that are not found in the claims themselves.”) (emphasis added) (citations and internal quotation marks omitted); *see also Dealertrack, Inc. v. Huber*, 674 F.3d 1315, 1322 (Fed. Cir. 2012) (“The disclosure of multiple examples does not necessarily mean that such list is exhaustive or that non-enumerated examples should be excluded.”).

¹⁷ However, neither Patent Owner’s Response nor Dr. Shenoy discusses whether the MYSQL database system mentioned in the ’707 patent requires database blocks with metadata. *See* Ex. 2017 ¶ 43, n. 11 (citing Ex. 1001, 5:63–67 and listing “MYSQL, and the like,” without addressing metadata in MYSQL).

In addition to the documents describing various commercially available database systems, Dr. Shenoy also discusses a treatise on database systems and testifies that it is generally understood that a database block or a page will include metadata. Ex. 2017 ¶ 40, n.8 (citing Ex. 2014 (“Molina”), 29. The portion of the Molina treatise cited by Dr. Shenoy, however, describes features of a *relational* database system, not characteristics common to all databases in general. See, e.g., Ex. 2014, 29 (“Collections such as *relations* are usually represented by placing the records that represent their data elements in one or more blocks.”) (emphasis added), see also *id.* at 31 (“Records representing tuples of a *relation* are stored in blocks of the disk [T]here is a *block header* holding information such as: . . . Information about which *relation* the tuples of this block belong to.”) (first and last emphasis added).¹⁸ Furthermore, rather than showing a normal meaning of a “database block,” the cited page shows that Molina refers to “one or more blocks,” “blocks of the disk,” and “a disk block,” for example, stating “it is normal for a disk block to hold only elements of one relation, although there are organizations where blocks hold tuples of several relations.” Ex. 2014, 29. In other words, similar to the ’077 patent Specification, as discussed further below, Molina does not support an alleged normal meaning of a “database block”—it obscures any alleged normal meaning of a database block because it does not use that term. See Pet. Reply 11 (arguing that all of Patent Owner’s related patents repeatedly interchange “data block” and “database block”).

¹⁸ Dr. Shenoy also cites to pages 20 and 21 of Exhibit 2014, but these pages correspond to the Table of Contents in Molina. See Ex. 2017 ¶ 40, n.8.

Petitioner contends that Patent Owner does not provide any evidence that database blocks require metadata in database systems other than relational database systems. *See* Pet. Reply 10. Petitioner correctly points out that the '077 patent Specification announces that the disclosed invention “can be used for *any database*.” *Id.* (quoting Ex. 1001, 6:6–7 (emphasis by Petitioner).) Relying on supplemental testimony of Dr. Zadok (Ex. 1063) submitted with its Reply, Petitioner asserts that, in other types of database systems, such as Google’s BigTable database, metadata is stored separately from the database blocks.¹⁹ Pet. Reply 10–11 (citing Ex. 1049, 4; Ex. 1063 ¶ 22). Petitioner also argues, citing the testimony of Dr. Zadok and another treatise on database systems (Ex. 1050, “Elmasri”), that a “flat file database,” such as a comma separated value (CSV) file used by spreadsheet applications, do not have metadata in its blocks. Pet. Reply 11 (citing Ex. 1063 ¶¶ 23–25, Ex. 1050, 431; Ex. 1053 (flat file database “is unlike a relational database” according to tekopedia.com); Ex. 1055, 15 (examples of flat files or ordinary text fields). Therefore, Petitioner asserts that the

¹⁹ Petitioner argues the definition of a “database” proposed by Dr. Shenoy, “a collection of data that is organized so that it can be easily accessed, managed or updated,” encompasses many types of databases other than the relational databases. *See* Pet. Reply 10–11 (citing Ex. 2017 ¶ 33). In addition to disclosing that “the techniques disclosed can be used for any database,” the '707 patent provides another broad definition— “[a] database comprises data stored in a computer for use by computer implemented applications.” Ex. 1001, 5:59–60, 6:6–7. Under either description or Mr. 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, spreadsheet databases, and “any database.”

ordinary meaning of database block does not require including metadata.
See id.; Ex. 1063 ¶ 25.²⁰

Petitioner also relies on Dr. Zadok’s testimony that contradicts Dr. Shenoy’s testimony in several aspects. Pet. Reply 9–10 (citing Ex. 2003, 271, 273; Ex. 1001, 14:42–49; Ex. 1063 ¶¶ 13–18). Dr. Zadok explains that at least Oracle and IBM DB2 also include empty database blocks *without* metadata, explains that Patent Owner relies on relational databases, and concludes that “[t]he ordinary meaning of the term ‘database block’ does not include metadata.” Ex. 1063 ¶¶ 13, 21, 25

We agree with Petitioner that the extrinsic evidence Patent Owner relies upon to argue for a limited ordinary meaning rests on relational databases. Because the challenged claims plainly recite “database,” and, therefore, are not limited to relational databases, Patent Owner’s evidence, even assuming it shows that most or all relational databases have database blocks that require metadata, does not establish an ordinary and customary meaning of “database blocks” recited in the claims. Therefore, we find Patent Owner’s extrinsic evidence regarding ordinary meaning does not overcome the intrinsic record of this case. *See Phillips*, 415 F.3d at 1318 (“[A] court should discount any expert testimony that is clearly at odds with the claim construction mandated by the claims themselves, the written description, and the prosecution history, in other words, with the written

²⁰ Similar to Exhibits 1020 and 1048 discussed above, Patent Owner characterizes cited Exhibits 1049, 1050, 1053, and 1055 as improperly submitted evidence, because they do not appear in the Petition. Paper 58, 1–2. Contrary to the characterization, we find this Exhibit evidence responsive to Patent Owner’s arguments regarding the claim construction of a database block.

record of the patent.”) (quotation marks omitted). Furthermore, we are persuaded by Petitioner’s evidence that certain well-known databases, including Google’s BigTable database and other databases, such as flat file databases, as discussed immediately above, do not require metadata in database blocks, and, therefore, find that the evidence in this case does not establish a widely accepted meaning of “database block” in the field of database systems that requires metadata to be necessarily included in a database block. *See supra*, note 19; Ex. 1063 ¶¶ 21–25 (discussing and citing Exs. 1050, 431–32; Ex. 1053; Ex. 1055, 15; Ex. 1049, 4).

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

F.3d at 1321 (“Properly viewed, the ‘ordinary meaning’ of a claim term is its meaning to the ordinary artisan after reading the entire patent.”).

ii) “*Application Level*” vs. “*Storage Level*”

Patent Owner argues, citing the passages in the ’707 patent describing certain aspects of the streaming embodiment, that a “database block” must include metadata because “the ’707 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), 13–14 (same or similar argument, citing Ex. 1001, 14:32–49, 15:15–19). A subsection heading announces that a “‘database block’ is not *storage level* unit of data.” *Id.* at 26 (original emphasis deleted, emphasis added). The passages cited by Patent Owner, however, describe processing in the streaming embodiment discussed above, and therefore, do not limit the challenged claims. *See* PO Resp. 13–14 (citing Ex. 1001, 14:32–49, 15:15–19. Furthermore, the passages related to streaming do not relate clearly to whether or not a “database block” is an “application level” or “storage level” entity or concept.

Patent Owner also asserts that the system of the ’707 patent “operates at the application level” because the system uses APIs (application program interfaces) to copy database blocks. *Id.* at 15 (citing Ex. 1001, 8:39–41). Petitioner argues that Patent Owner’s argument is contrary to disclosed embodiments of the ’707 patent because the patent teaches the system “may retrieve the necessary database blocks from *storage level snapshots* of production databases.” Pet. Reply 13 (citing Ex. 1001, 9:24–26 (emphasis by Petitioner)). We agree with Petitioner’s argument and further note that

the '707 patent also discloses copying of database blocks from *storage level* snapshots of a database. See Ex. 1001, 7:4–10 (“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 a disclosed API embodiment 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,” “data blocks,” “blocks of data,” or “storage level” unit of data. PO Resp. 23–28, n.9 (citing Ex. 2017 ¶¶ 87, 104, 107, 128, 132). In particular, Patent Owner asserts that an Oracle document (Ex. 2003, “Oracle Manual”) explains this “well-known distinction.” *Id.* at 26–27 (citing Ex. 2003, 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 29 (citing Ex. 2017 ¶¶ 121–34). Therefore, Patent Owner argues a construction of the term “database block” that encompasses a storage unit block is incorrect. *Id.* at 29–30.

As a threshold matter, we note that the Oracle Manual document is dated May, 2014, which is almost five years later than the filing date of the

'077 patent, October, 2009. *Compare* Ex. 2003, 1, *with* Ex. 1001, 1. 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, the '707 patent Specification 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 '707 patent 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.*, 582 F.3d at 1348; *Phillips*, 415 F.3d at 1318.

iii) Specific Number of Bytes in a Database Block

In the Institution Decision, we found that database blocks of the '707 patent need not have a specific number of bytes stored in storage because an empty database block need not have “a specific number of bytes stored in storage.” Dec. on Inst. 8. Citing passages in the '808 patent, Patent Owner contends that “[t]he correct interpretation of the term ‘database block’ 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.’” PO Resp. 17 (citing Ex. 2001, 2:7–12). Again, Patent Owner does not direct attention to a comparable disclosure in the '077 patent. *See id.* at 17–18.

Even if one exists, Patent Owner does not explain how to interpret “a specific number of bytes stored in storage.” For example, it is not clear if “a specific number” is a constant across all databases. In another argument, Patent Owner states that “[d]atabase blocks may be implemented as any number of different sizes independently of the size of the file system data blocks which may ultimately store the database data.” PO Resp. 29 (citing Ex. 2017 ¶¶ 121–34). 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 28 (citing Ex. 2017 ¶ 128).

According to Patent Owner’s arguments and the record, as noted above, a database block may be “any number of different sizes.” PO Resp. 29. Therefore, the database blocks of the ’077 patent need not have a constant specific number of bytes stored in storage.

iv) “Database Block” vs. “File” Distinction

Next, Patent Owner argues that the construction proposed by Petitioner (Pet. 10) and preliminarily adopted in our Institution Decision (Dec. on Inst. 11)—“a unit of data used by a database”—is improper because it would equate a “database block” with a “file.” PO Resp. 25–26. Citing dictionaries, Patent Owner argues that the ordinary meaning of “file” is “a unit of data.” *Id.* at 25 (citing Ex. 2020, 3; Ex. 2021, 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 a database.” *Id.* at 25–26.

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. Pet. Reply 13 (citing Ex. 1031, 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 omitted when quoting the sources in the Patent Owner Response, show that a file name is a required attribute of a file. See Ex. 2020, 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. 2021, 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.

v) Database Blocks, Blocks of Data, and Data

Patent Owner contends that the '077 patent Specification does not use the term “database blocks” and “blocks of data” interchangeably. PO Resp.

20. Patent Owner contends that the '077 patent only uses “blocks” in the context of “blocks of virtual database files that point to database blocks” or uses the term “database block” and “then uses the word ‘block’ as shorthand to refer to these database blocks.” *Id.* at 21. Petitioner disagrees, and argues that the '077 patent uses database blocks and generic blocks of data interchangeably. Pet. Reply 11–12 (citing Ex. 1001, 11:6–10; Ex. 1063 ¶¶ 26–29; Ex. 1031, 251:20–255:21–22).

The record supports Petitioner and does not support Patent Owner, even if the “shorthand” use does not obscure any supposed distinction. For example, in one sentence, the '077 patent *introduces* “a block B1” and “a block B2,” and *then* refers to those blocks as “database block B1” and “database block B2.” Ex. 1001, 15:50–55. As another example, the '077 patent Specification describes “sav[ing] blocks of data,” keeping track of “each block of data,” “making copies of blocks of data,” and “updating the original block of data.” *See* Ex. 1001, 11:7–21. The '077 patent also refers to “the latest copy of the blocks F33 and F34,” “shar[ing] block F11 with block U11 of VDB files,” and announces that “the sharing of blocks across multiple VDBs results in efficient[] utilization of data stored in the storage system data store.” *Id.* at 19:57–64. The '077 patent Specification also refers generically to “data” multiple times in the context of the invention. *Id.* at 23:45, 24:4, 24:31, 24:26, 24:57, 24:65, 27:2–3, 27:31, 27:42, 28:59.

Patent Owner admits that “‘data blocks’ refer to blocks of data stored on a disk and manipulated by the file system.” PO Resp. at 27, n.9; Pet. Reply 12. Accordingly, contrary to Patent Owner’s arguments, similar to the extrinsic evidence discussed above with respect to Molina (Ex 2014;

supra Section C.1.i), the '077 patent Specification does not distinguish between blocks of data, data, and database blocks. This implies that the terms have similar meanings in light of the '077 patent Specification.

vi) *Conclusion*

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

2. *Virtual Database*

Challenged independent claim 1 recites “[a] method . . . using a *virtual database* system, the method comprising . . . linking, . . . loading . . . and provisioning . . . a development *virtual database* . . . and a test *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.” Dec. on Inst. 15. Petitioner agrees with this construction. Pet. Reply 13–14. Patent Owner, on the other hand, asserts

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.” PO Resp. 30. Patent Owner relies on the following sentence in the ’077 patent Specification to support the “decoupled” aspect of its construction: “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.” *Id.* (quoting Ex. 1001, 6:10–13). Patent Owner relies on the following sentence to support the “database server” aspect: “The set of files associated with the VDB are mounted on a database server allowing the database server to read from and write to the set of files.” *Id.* (quoting Ex. 1001, Abstract).

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

i) Whether Reading and Writing by “a Database Server” Is Necessary

Patent Owner argues that the point of the first proposed phrase is to “require[] something of the files . . . that they are of a form readable and writable by a database server.” PO Resp. 33–34. 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 into the proposed definition. *See id.* at 34.

Challenged independent claims 1 and 18 specifically recite “mounting the set of files to the system to allow a database server running on the system to access the set of files.” Claim 1 of the related ’808 patent more narrowly recites “allowing the database server to read from and write to the set of files.” Hence, Patent Owner could have claimed the limitation in a server and chose not to do it. *See LSI Indus., Inc. v. ImagePoint, Inc.*, 279 F. App’x 964, 972 (Fed. Cir. 2008) (rejecting the district court’s construction of “display device” as necessarily including the superfluous limitation of “internal illumination” because other claim terms specifically recited an “illuminated display device”); *but cf. ERBE Elektromedizin GmbH v. Canady Tech. LLC*, 629 F.3d 1278, 1286 (Fed. Cir. 2010) (“no canon of [claim] construction is absolute in its application”).

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.” *See* PO Resp. 34 (citing Ex. 2017 ¶ 99). Patent Owner’s argument is unpersuasive because our construction already requires a virtual database to include “a set of database files.” The ’707 patent tracks our construction and describes that a virtual database comprises “database files.” *See, e.g.,* Ex. 1001, 6:10–13 (“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.”) (emphasis added), 6:31–32 (“A virtual database may be created on a database server by creating the *database files*.”) (emphasis added).

Patent Owner also argues that “[t]o be a database file, a database server must be able to read, write and understand its contents.” PO Resp. 34 (citing Ex. 2017 ¶ 99). As noted above, claim 1 specifies that a database server accesses the set of files. The ’707 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, 5:59–60. Furthermore, the ’077 patent Specification announces that the disclosed invention “can be used for *any database*.” *Id.* (emphasis added). Furthermore, “[a] database server is a computer program that can interact with the database and provides database services, for example, access to the data stored in the database.” *Id.* at 5:60–63. During the hearing, Patent Owner clarified that that a database server in its proposed construction “is used synonymously . . . with database application.” Tr. 185:11–12. In line with Patent Owner’s acknowledgement at the oral hearing, the ’077 patent, and the meaning of a database, database files have the capability of being read from, and written to, by a computer program or application.

The Specification shows that the readable and writable characteristics are inherent attributes of a virtual database. *See, e.g.*, Ex. 1001, 19:15–16 (“FIG. 10 indicates how storage efficient copies are made to create a *read/write file structure representing a VDB*.”) (emphasis added); 19:45–51 (“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); 18:67–19:3 (“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.

Nevertheless, the phrase “database files” need not be further modified to indicate explicitly that the files are capable of being read to, and written from, by a computer application or program (i.e., a database server), because, as discussed above, the “database” description provided in the Specification (see *supra* note 19), and the recited database server in the challenged claims, already requires a database server to be able to access the database files. 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.”

*ii) “Decoupled from the Logical Use of the Database Files
by the Database Server”*

Patent Owner relies on the following sentence in the ’707 patent to support the second part, i.e., the “decoupled” aspect, of its proposed construction: “virtual databases are ‘virtual’ in the sense that the physical implementation of the database files is decoupled from the logical use of the database files by a database server.” PO Resp. 30 (quoting Ex. 1001, 6:10–13). Patent Owner asserts this sentence explicitly defines the use of the word “virtual” in the term “virtual database.” *Id.* However, the ’707 patent does not describe clearly 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 ’707 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. 2017 ¶¶ 23–26, 63–70, 87–90). Petitioner asserts Patent Owner confuses “platform virtualization” with “database virtualization” or “virtualizing data.” *See* Pet. Reply 16 (citing Ex. 1063 ¶ 37). We agree with Petitioner that “platform virtualization” is unrelated to the concept of “virtual database” as claimed and described in the ’707 patent.

In his Declaration, Dr. Shenoy testifies that the Java programming language running on a Java Virtual Machine (JVM) as described in an Oracle document explains the general concept of platform virtualization. *See* Ex. 2017 ¶¶ 23, 24 n.6 (citing Ex. 2002, 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” (Ex. 2017 ¶ 23) is achieved by the Java Virtual Machine, which “is *software* that emulates the operation and interface of a physical processor” (*id.* at ¶ 24 (emphasis added)). The ’707 patent also describes that a “virtual machine” is “provided by *platform virtualization software*” or “server virtualization *software*.” Ex. 1001, 8:9–12 (emphases added). Further, the ’707 patent describes a Java Virtual Machine as part of production database system 110 (Ex. 1001, 9:4–7), which is separate from virtual database 220 stored on database storage system 100, as discussed above in Section II.A. *See* Ex. 1001, Fig. 2(a). Hence, Patent Owner’s discussion of the Java Virtual Machine may relate to database servers or other *software* of the disclosed system, but it is *not* directed to a virtual

database, which is a set of *files* as expressly recited in the claims and described in the Specification. *See, e.g.*, Ex. 1001, 6:23–26, Figs. 2a, 10, 12, Abstract.

Other examples of “decoupling” that 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 ’707 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. 2017 ¶¶ 69–70 (citing Ex. 1001, 8:39–44). The latter cited portion of the ’707 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, Fig. 3, 8:39–44 (“An example of a vendor interface module is the *program code* of a database server provided by vendor ORACLE that implements RMAN APIs.”), 14:1–3 (“the production system library 385 includes *code* to analyze the structures of the files of the database stored in the data store 350”) (emphases added).

Dr. Shenoy and Patent Owner also describe and rely upon virtual database systems 130c depicted in Figure 1 of the ’707 patent. PO Resp. 31 (citing Ex. 2017 ¶ 92). However, Figure 1 shows 130c as the “Virtual Database System 130(c),” instead of the disclosed (and claimed) virtual database, which is stored in a different system, namely, Database Storage System 100. *See* Ex. 1001, Figs. 1, 2a, 2b. That is, virtual database *system* 130 is separate from and external to *virtual database* 220. *Id.* Further, the

'707 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, 12:61–13:4 (“A virtual database system 130 includes a database server 360 and a VDB system library 380 . . . The VDB system library 380 contains *program code* for processing requests sent by the database storage system 100.” (emphasis added)); Fig. 3. Hence, Patent Owner appears to conflate the function of the *software* of a virtual database *system* with the meaning of “a virtual database,” which is a set of *files* that resides in a database storage system that is separate from the virtual database system.

Petitioner argues that Patent Owner’s platform virtualization argument is misguided because it conflates the meaning of a “virtual database” with the disclosed embodiments of production database *system* and the virtual database *system*. Pet. Reply 16–17. We agree with Petitioner as discussed above. Patent Owner’s arguments are directed to the software part of the disclosed system that is separate from and external to a virtual database. As described in the '707 patent, a database system generally comprises a database, which is data stored in storage, and database software, such as database servers or other programs that access the database. *See* Ex. 1001, 5:59–67; 8:27–30. 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. 2017 ¶ 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 set of files pointing to and including the *data part* (database blocks) of a database system.²¹ Therefore, contrary to Patent Owner’s argument, virtualization of computing platform or storage by software is unrelated to and cannot be equated to virtualization of the data files and 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 the Patentees intended to claim the function or operation of the software of the disclosed

²¹ “Each [decoupled database] file *includes* a set of database blocks and the data structures for referring to the database blocks.” Ex. 1001, 6:25–26 (emphasis added). In other words, apparently, the database files “include” the “database blocks” in the sense that the files point to (i.e., are linked with) the database blocks (also, as noted, a different set of files may point to the same database blocks), so that the use of the files (writing thereto and reading therefrom) is decoupled from the location (including the storage of data in) the database blocks. See Ex. 1001, Figs. 10–12; Ex. 1009 ¶¶ 38–45 (describing the virtual database pointing system of the ’077 patent).

system, they 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.

Nevertheless, 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. 33 (citing Inst. Dec. 17). Patent Owner’s argument is unpersuasive. As discussed above and below, our claim construction is consistent with the language of the claims and the written description of the ’707 patent, and implicitly includes the concept of virtualization, without including every feature that Patent Owner argues the concept requires.

Challenged independent claim 1 recites

[a] method . . . using a *virtual database* system, the method comprising: *linking* a source database . . . provisioning a development *virtual database* to a development system and a test *virtual database* to a test system, wherein provisioning each of the *virtual databases* to a system comprises: *creating a set of files linked* to the stored database blocks on the storage system.

Ex. 1001, 35:27–43 (emphasis added).

Tracking the language of claim 1, the ’077 patent Specification describes that “[t]he provisioning of the virtual databases is performed by creating a set of files *linked* to the stored database blocks on the storage system.” Ex. 1001, 2:11–12 (emphasis added). Similarly, “[a] virtual

database (VDB) is created by creating a set of files in the data storage system. Each file in the set of files created for a VDB is linked to the database blocks on the database storage system associated with a point-in-time copy of the source database.” *Id.* at Abstract. Hence, the database files of a VDB are linked or mapped to the database blocks associated with the database being virtualized.

Looking to the ’077 patent Specification for further clarification, Figure 10 of the ’707 patent depicts “how . . . to create a read/write file structure representing a VDB.” *Id.* at 19:15–16. 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 19:34–37 (emphasis added). The ’707 patent further describes this pointing system as follows:

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 19:37–43 (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 F1, F3, etc. stored in a physical storage device.

Another disclosed example shows 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 20:39–45. The ’707 patent further describes “[s]ince the [virtual database file] structure 1050 illustrated in FIG. 10, structure 1150 illustrated in FIG. 11, or structure 1350 illustrated in FIG. 13 are read/write structures, the virtual database system 130 is allowed to read from these structures as well as write to them.” *Id.* at col. 20:33–36. As described further, “[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 20:52–55.

Thus, the ’707 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 VDB files, but the system actually writes the data elsewhere to another physical location specified by the pointing blocks such as V11 and V12. *See id.* at 20:32–55, Figs. 10–12. They also are virtual in the sense that database blocks may be shared by a different set of files corresponding to different virtual databases. *See id.* at 20:52–55.

Based on the disclosure in the ’707 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 ’707 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.

Dr. Shenoy acknowledges that Figure 10 illustrates the “decoupling of the physical storage of the database from its logical use,” in part, because “common database blocks are shared between different point-in-time copies. This is imperceptible to the user.” Ex. 2017 ¶ 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.* ¶ 97 (emphasis added).

Nonetheless, Patent Owner argues that the preliminary construction is erroneous because “*any* database, including a non-virtual database, can be mapped to physical addresses for stored database blocks.” PO Resp. 33 (citing Ex. 2017 ¶ 100). Patent Owner also explains that the construction “would cover a non-virtual data base” and cover “an exact copy of a source database.” *Id.* at 32 (citing Ex. 2017 ¶¶ 95–96).

Patent Owner’s arguments are partially persuasive. Accordingly, we modify our preliminary construction. 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.”

In other words, the construction now requires the stored database blocks to be associated with *another* database. As Patent Owner’s

arguments imply, an ordinary (i.e., a non-virtual database) consists of its own database blocks and would *not* be created by mapping or pointing to stored blocks associated with *another* database. Furthermore, the '707 patent describes that a virtual database may be a source database (or vice versa). See Ex. 1001, 2:9–11; 20:5–6; *see also* Ex. 2001, 35:1–3 (claim 6 of the '808 patent reciting “wherein . . . the source database is a second virtual database”).

The claim construction tracks the language of the challenged claims, the '077 patent Specification, Dr. Zadok's testimony, and Dr. Shenoy's testimony, all of which show how to create a VDB (i.e., using a set of files), as discussed above. See Ex. 2017 ¶¶ 63, 97; Ex. 1001, Abstract.

iii) Conclusion

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.”

D. Obviousness

1. Principles of Law

A claim is unpatentable under § 103(a) if the differences between the claimed subject matter and the prior art are such that the subject matter, as a whole, would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). The question of obviousness is resolved on the basis of the following underlying factual determinations: (1) the scope and content of the prior art; (2) any

differences between the claimed subject matter and the prior art; (3) the level of skill in the art; and (4) where in evidence, so-called secondary considerations. *Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966).

2. *Sanders*

Sanders describes a system “to clone a DB2 database quickly and easily.” Ex. 1013, 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, “[a] FlexClone volume is a writable point-in-time image of a FlexVol volume or another FlexClone volume.” *Id.* (emphasis added). 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 has all the features and capabilities of a regular FlexVol volume, including . . . the ability to be the source of another FlexClone volume.” *Id.* at 3. Thus a FlexClone volume can be based off a PIT copy of another FlexClone volume. *See id.* (“A FlexClone volume is a writable point-in-time image of a FlexVol volume or another FlexClone volume.”).

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 “an efficient way to transfer data” as read-only copies at the same or at different destinations. *Id.* at 3. “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.*

Sanders’s Figure 3, annotated by Petitioner to add source and clone database labels, follows:

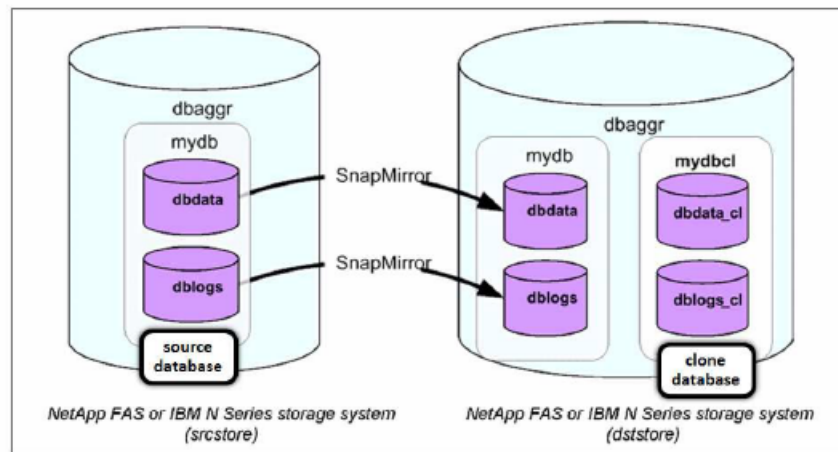


Figure 3) Clone database volumes on the second NetApp FAS or IBM N Series storage system

Sanders at Fig. 3 (annotated)

Pet. 19 (annotating Ex. 1013, Fig. 3). Figure 3 represents using the combined SnapMirror and FlexClone technology to transmit PIT clone copies of a production database to a destination storage system. See Ex. 1013, 8.

Sanders also describes mounting a clone database on a database server. See *id.* at 14 (“In order to access the clone database, you need to mount the clone volumes to a database server. . . . [Y]ou can mount the

clone volume by executing the following command on the database server:
mount [MountPoint].”).

3. Edwards

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

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

Ex. 1014, 11 (second emphasis added). Edwards describes the same underlying SnapMirror, FlexVol, and FlexClone systems that Sanders describes, as forming a virtualized system. *See, e.g., id.* at 10 (describing “Snapshot copies” and flexible volumes introduced by “NetApp in release 7.0 of Data ONTAP®”), 13 (“Volume SnapMirror”), 14–15 (“FlexVol,” “FlexClone”). SnapMirror mirrors the contents of a volume from a source system to a destination storage system. *Id.* at 13.

Using these systems, “we *virtualize* the allocation of volumes on physical storage, allowing multiple, independently managed file volumes, along with their Snapshot copies, to share the same storage.” *Id.* at 10 (emphasis added). “*Virtualization is a well-known method of abstracting physical resources and of separating the manipulation and use of logical*

resources from their underlying implementation.” Id., Abstract (emphasis added). “*The resulting virtual file volumes, or FlexVol® volumes, are managed independent of lower storage layers.” Id.* (emphasis added).

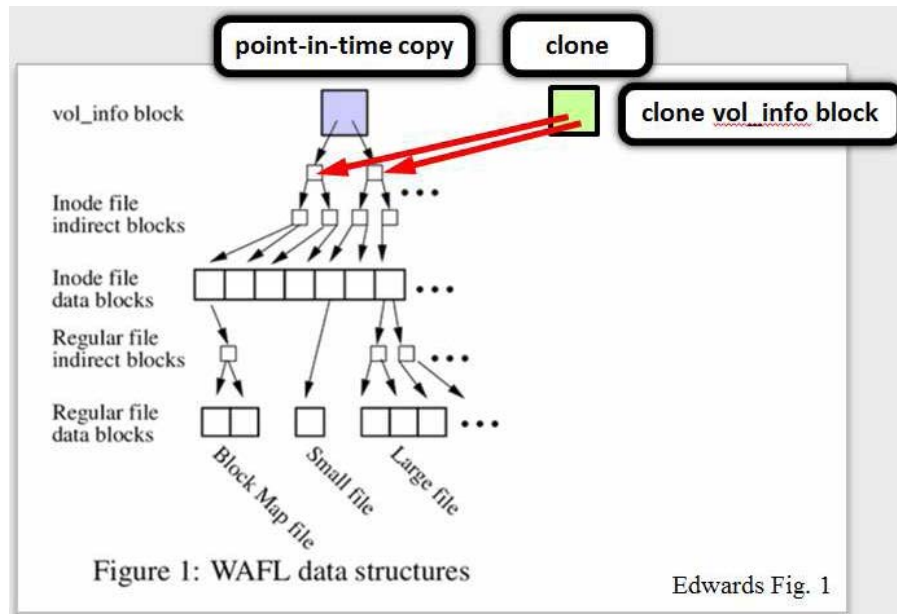
Edwards also describes mapping virtual blocks to the addresses of physical blocks: “*Mapping between virtual block addresses used by FlexVols and physical block addresses” Id.* at 10 (emphases added).

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

Edwards explains that “[c]reating a clone volume is a simple process.” *Id.* at 15. A container file for the new clone volume (i.e., a FlexClone volume) is created and seeded “with a vol_info block that is a copy of the vol_info block of the Snapshot copy on which the clone is based.” *Id.* Because vol_info block is the root of the “tree of blocks that form the

Snapshot copy, the clone inherits pointers to the complete file system image stored in the original snapshot copy.” *Id.*

Petitioner provides an annotated version of Edwards’s Figure 1, to demonstrate this step in cloning, as follows:



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

4. Alleged Obviousness Ground: Sanders and Edwards

i) Claims 1, 3, 5–7, 9, 10, 18, 20–23, 25, and 26

Relying on its declarant, Dr. Zadok (Ex. 1009), Petitioner asserts that the combination of Sanders and Edwards renders obvious claims 1, 3, 5–7, 9, 10, 18, 20–23, 25, and 26. Pet. 30–53. Petitioner points to specific disclosures in the prior art that are deemed to describe, teach, or render obvious all the claim limitations. *See id.*

Addressing the preamble of claim 1, Petitioner contends that, even if it is afforded weight, Sanders’s FlexVol, SnapMirror, and FlexClone

systems together create writable cloned volumes as a virtual database system. Pet. 15–25, 30–31 (citing Ex. 1009 ¶¶ 122–124). According further to Petitioner, Edwards’s system, which describes the same underlying NetApp system that Sanders describes, has a physical implementation of the files in the clone volume decoupled from the logical use of the files, creating a virtual system, because the files “‘point[]’ to the pre-existing data blocks associated with a point-in-time copy of its parent volume instead of physically duplicating the data blocks of its parent.” See Pet. 31 (citing Ex. 1009 ¶ 124); Pet. 25 (quoting Ex. 1014, 141 and stating that FlexClone volumes “‘separate the management of physical storage devices from the management of logical data sets’”).

The first listed step of claim 1 recites “linking a source database, wherein linking the source database comprises receiving information identifying the source database.” Petitioner contends that Sanders teaches this step by executing a “snapmirror initialize” command at a destination storage system to identify a source IBM DB2 database volume for subsequent cloning. Pet. 31–32 (citing Ex. 1013, 22–23; Ex. 1009 ¶ 127).

The next listed step of claim 1 requires loading multiple PIT copies: “loading a plurality of point-in-time copies of the source database, wherein the loading comprises; receiving database blocks for the point-in-time copies of the source database, and storing the database blocks on a storage system.” Petitioner contends that Sanders’s SnapMirror system satisfies this step, quoting Sanders, as follows: “As noted earlier, the initialization process ***transfers data***, including ***all Snapshot copies*** from the source volume ***to the destination volume*** for the first time; thereafter, only changed blocks are

transferred.” Pet. 33 (quoting Ex. 1013, 21(emphases by Petitioner); citing Ex. 1009 ¶ 128).

The next listed step requires provisioning development and test virtual database systems. Petitioner points out that the ’077 patent Specification describes provisioning as making the virtual database available to a database server. Pet. 35 (citing Ex. 1001, 6:38–40).

Petitioner relies on Sanders’s teaching of “access[ing] the clone database . . . [by] mount[ing] the clone volumes to a database server.” Pet. 35 (quoting Ex. 1013, 28). Petitioner also quotes Sanders as follows: “[a] database clone can be created simply by creating a clone of the FlexVol volumes that are used for the database’s data and transaction logs.” Pet. 35 (quoting Ex. 1013, 3). Petitioner further explains that

[i]n addition, Sanders discloses provisioning both a development and test FlexClone volumes. “Database **cloning** is performed **frequently** by database administrators to provide near-production data for various business needs such as **application development, QA testing**, and report generation.” See Sanders at 3; Zadok Decl. at ¶ 135. Given that “cloning is performed frequently,” (*id.*) a POSITA would understand that separate test and development clones can be created and mounted. See *id.*

Pet. 35 (emphases by Petitioner).

Petitioner relies on similar teachings in Edwards that describe making clones that share blocks with a base snapshot and making writable copies for “development or test purposes.” See Pet. 35–36 (citing Ex. 1009 ¶ 135; Ex. 1014, 15; quoting Ex. 1014, 14).

The two final listed claim steps require “creating a set of files linked to the stored database blocks on the storage system” and “mounting the set

of files to the system to allow a database server running on the system to access the set of files.”²²

Sanders’s system mounts the clone volume to a server in order to access the data, as noted above. *See* Ex. 1013, 28. Petitioner points out that Sanders discloses that “[a] database clone can be created simply by creating a clone of the FlexVol volumes that are used for the database’s data and transaction logs.” Pet. 36 (citing Ex. 1013, 3).

Addressing linking, Petitioner relies partly on the following teachings in Sanders:

The clone database that results “is a frozen image of the database file system at the time of the clone creation.” [Ex. 1013, 6]. “[T]he original FlexVol volume and the FlexClone volume [] share common data.” *Id.* at 3. Sanders discloses creating a FlexClone volume which shares—or is “linked” to—data blocks of a point-in-time copy of a source database. [Ex. 1009 ¶ 139].

Pet. 36.

Petitioner contends that although Sanders does not set forth explicitly all the details for creating the linking files, Edwards, which describes the same underlying NetApp system, “provides 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.” Pet. 37.

Relying on Edwards, and Dr. Zadok, according to Petitioner,

[a] container file for the new clone volume is created and seeded “with a vol_info block that is a copy of the vol_info

²² The “database server running on the system” inferentially refers back to one or more systems having antecedent basis in claim 1: “a virtual database system,” “a system,” “a storage system,” and “the storage system.”

block of the Snapshot copy on which the clone is based.” [Ex. 1014, 15.] Thus, the cloned volume inherits all the pointers or links of the Snapshot copy that point to the underlying data blocks. Thus, the clone volume “***inherits pointers to the complete file system image stored in the original Snapshot copy.***” [Id.]. By creating a new vol_info block that points or links to the tree of blocks of the Snapshot (point-in-time) copy, Edwards discloses creating a new, cloned volume comprising a set of database files, by pointing to already-stored database blocks associated with a point-in-time copy of the source database. [Ex. 1009 ¶¶ 140–142].

Pet. 37 (emphasis by Petitioner)).

ii) Reasons for Combining Sanders and Edwards

Petitioner contends that combining Sanders and Edwards, which disclose “the same NetApp technology and features, albeit in varying levels of detail,” would have been obvious in order to gain a more complete understanding of the system. Pet. 26–27 (citing *Ex Parte Mettke*, No. 2008-0610, 2008 WL 4448201 (BPAI Sept. 30, 2008), *aff’d*, 570 F.3d 1356 (Fed. Cir. 2009) (Ex. 1011); Ex. 1009 ¶ 86). According to Petitioner, NetApp introduced a number of database products for cloning:

All of these products integrated the SnapMirror, FlexVol, and FlexClone technologies described in Sanders and Edwards. While NetApp did not introduce a similar, stand-alone product dedicated for DB2 databases, a POSITA would have readily referred to Sanders to apply the cloning technique described in Edwards to clone DB2 databases. [Ex. 1009 ¶ 99].

Pet. 29.

A POSITA would have been motivated to combine Sanders and Edwards, given that they recite the same FlexClone, SnapMirror, and Snapshot features from the DataONTAP platform by NetApp, and a POSITA would have looked to Edwards to understand the specifics of the FlexClone

creation process of how FlexClone creates files linked to stored database blocks on the storage system. [Ex. 1009 ¶ 143].

Pet. 39.

Petitioner provides other and similar reasons for combining the various aspects of the NetApp technologies at various places in the Petition. *See* Pet. 26–29, 31, 34, 36. In summary, Petitioner asserts that combining the teachings of the same underlying integrated cloning systems from the same corporation as described in Sanders and Edwards would have provided an efficient mechanism for cloning databases in order to facilitate testing or data back-up, something desired by companies for managing data. *See id.*

iii) Summary of Petitions

Petitioner applies the teachings of Sanders and Edwards in an asserted ground of obviousness as to claims 1, 3, 5–7, 9, 10, 18, 20–23, 25, and 26. Pet. 30–53. Petitioner applies the teachings of Sanders, Edwards, and Patel, in an asserted ground of obviousness as to claims 2, 4, and 8. Pet. 53–57. Petitioner applies the teachings of Sanders, Edwards, and Data ONTAP Guide, in an asserted ground of obviousness as to 8, 11–13, 24, and 27–29. ’026 Pet. 33–57. Petitioner applies the teachings of Sanders, Edwards, Data ONTAP Guide, and Sarma, in an asserted ground of obviousness as to claims 17 and 33. ’026 Pet. 57–60.

We find that the two Petitions provide ample reasons why one of ordinary skill in the art would have combined the teachings of Sanders, Edwards, Patel, and Data ONTAP Guide as set forth in the Petitions. *See* Pet. 15–30; ’026 Pet. 25–33. Sanders, Edwards, Patel, and Data ONTAP Guide teach cloning (i.e., virtual) file systems using the same underlying NetApp technologies: Snapshot, SnapMirror, and FlexClone. They each

employ the same solution, using cloning file systems, in order to solve the problem of avoiding making physical copies of data and creating efficient reductions in storage space, resources, cost, and speed for data retention and recovery. *See* Pet. 27–30; ’026 Pet. 26–33; Ex. 1013, 1, 3; Ex. 1014, 9–11, 14; Ex. 1016, 1–6; Ex. 1009 ¶¶ 90, 57–101; ’026 Ex. 1117, 7–9, 92; ’026 Ex. 1109 ¶¶ 56–117.

In addition, Dr. Zadok testifies that Sanders expressly refers to Data ONTAP Guide. ’026 Ex. 1109 ¶ 109 (citing ’026 Ex. 1113, 50 (i.e., a copy of Ex. 1013, 50)). Sarma suggests compression for storage efficiency and cites to systems including ONTAP and SnapMirror. ’026 Pet. 31 (citing ’026 Ex 1118, 7:11–14, 14:11–13; ’026 Ex. 1109 ¶ 113). The record supports, and we adopt, Petitioner’s contentions as summarized above.

iv) Patent Owner’s Response

Patent Owner groups together independent claims 1, 11, 18, and 27 and focuses arguments on claim 1. *See* PO Resp. 41–42; ’026 PO Resp. 40–41.

a. Motivation

Patent Owner argues that Sanders, Edwards, and Patel, do not describe the “same” technology, and that “Petitioners sole argument as to why the documents collectively render a claim limitation obvious is because, Petitioner maintains, they describe ‘the same’ technology.” PO Resp. 6. Patent Owner contends that “FlexVol, FlexClone and SnapMirror are NetApp trademarks [and products] . . . associated with products and technologies that changed over time,” showing that the documents describe different embodiments. *Id.*

On the other hand, Patent Owner concedes that “Petitioner offers other general reasons why one skilled in the art would purportedly combine the teachings of Sanders, Edwards and Patel.” *Id* at 5. Patent Owner makes similar arguments in the ’026 IPR with respect to Sanders, Edwards, Data ONTAP Guide, and Sarma. ’026 PO Resp. 5–7, 34–55, 40.

As Petitioner contends, and as we find above, with the exception of Sarma (which cites and applies to the NetApp systems as noted below), the documents focus on describing the same basic NetApp system, including FlexVol, FlexClone, and SnapMirror, all to provide database cloning, eliminating the need for physical copying, and providing savings in storage space, cost, data recovery, etc. *See* Pet. 15–26; Ex. 1009 ¶¶ 57–100; ’026 Pet. 26–33; Ex. 1109 ¶¶ 90–93, 56–117. Even if the products did change over time, as Patent Owner contends, understanding the evolution of the underlying related systems reasonably would have been beneficial to better understand the products and apply them for cloning solutions, data back-up, etc. *See, e.g.*, Ex. 1009 ¶ 100 (“Sanders, Edwards, and Patel describe the natural evolution of these technologies at NetApp”) ¶ 99 (starting with Sanders for cloning virtual databases, “the most obvious place to start would be to look at other NetApp publications”) ¶¶ 90–94 (Sanders, Edwards, and Patel address the same problem and solution and provide motivation for combining—e.g., creating multiple virtual databases copies, and providing writable virtual copies, in order to avoid the cost and complexity in making physical copies and to provide data for experimental, test, developmental purposes); ’026 Ex. 1109 ¶¶ 96–112 (describing Sanders and Edwards in the same fashion, adding that Sanders refers to Data ONTAP Guide, and that

Data ONTAP Guide, like Sanders and Edwards, provides additional features for making PIT using SnapMirror and FlexClone).

In summary, notwithstanding Patent Owner's arguments to the contrary, as we find above, Petitioner provides ample motivation and reasons to combine Sanders, Edwards, Patel, and Data ONTAP Guide.

b. Database Blocks, Database, Server

Patent Owner contends that the “*data blocks* [in the NetApp system of Sanders and Edwards] are not *database blocks*,” because “NetApp’s storage technology is based on its proprietary file system called WAFL, which operates on physical storage systems and organizes data to be stored.” PO Resp. 37. Patent Owner contends that “Sanders does not address *database blocks* or *database files*.” *Id.* at 40–41 (citing Ex. 2017 ¶ 116). Patent Owner argues that the combined system fails to render obvious *database blocks*. *Id.* at 43–44. According to Patent Owner, in the NetApp system, “when an application asks to read a file, the file system looks up the proper inode for the file and traverses its structure to find the appropriate *data blocks* which store the file content.” *Id.* at 38 (citing Ex. 2017 ¶ 106) (emphasis added). According further to Patent Owner, “[a]t a minimum, a ‘*database block*’ must be a unit of data recognized and used by a database server application. Indeed, the patent repeatedly describes that *database blocks* are accessed and obtained through Application Programming Interfaces (APIs).” *Id.* at 46 (emphasis added).

According further to Patent Owner:

Petitioner attempts to draw a parallel between WAFL’s use of “inodes” and pointers to *data blocks on disk* with the claimed virtual database system and its use of *database blocks* and files

linked to the database blocks. The NetApp documents relied-upon by Petitioner, however, do not *virtualize* databases as described and claimed in the '077 patent, they describe storage level systems and technologies, not *database systems* and *database blocks*.

Id. at 39 (emphases added).

Patent Owner also argues that the NetApp snapshot copies do not constitute the PIT copies required by challenged claims 1 and 18. Patent Owner explains that “Snapshot copies do not include *database blocks*, they include file system *data blocks*.” *Id.* at 50 (emphases added). Similarly, Patent Owner contends that the inode database files in WAFL “are not capable of being mapped to physical addresses for stored *database blocks*, as required by the Board’s preliminary construction.” *Id.* at 54. Rather, Patent Owner contends that the “inode files are collections of inodes which may point to file system *data blocks*.” *Id.* Patent Owner presents the same claim construction and makes materially the same arguments in the '026 IPR. *See* '026 PO Resp. 17–59.

Patent Owner’s arguments primarily turn on the claim construction of “database block” and fail to rebut Petitioner’s showing. As described above, the '077 patent “techniques disclosed can be used for *any* database.” Ex. 1001, 6:5–6. The '707 patent provides another broad definition—“[a] database comprises data stored in a computer for use by computer implemented applications.” Ex. 1001, 5:59–60, 6:6–7. As Petitioner notes, Dr. Shenoy’s testimony shows that a database, as described in the '077 patent, encompasses a wide variety of databases. *See* Pet. Reply 10–11 (citing Ex. 2017 ¶ 33); *supra* note 19. As discussed above, the challenged

'077 patent claims and the '077 patent Specification do not require metadata in database blocks, do not require APIs, do not distinguish between database blocks and blocks of data, and do not preclude file systems or file system data blocks. *See supra* Section II.C.1 (Claim Construction of database block). In other words, the challenged claims read on the NetApp WAFL system as taught by the combination of Sanders and Edwards, which renders obvious the challenged claims, as set forth by Petitioner.

Further regarding arguments alleging the lack of a database system and database server, claim 1 requires “mounting the set of files to the system to allow a database server running on the system to access the set of files.” Petitioner persuasively shows that the NetApp database server recognizes the set of files of a database. For example, Petitioner relies on Sanders’s teaching of “access[ing] the clone database . . . [by] mount[ing] the clone volumes to a database server.” Pet. 35 (quoting Ex. 1013, 28); *see also* Pet. 19 (similar reliance on Sanders’s server, citing Ex. 1013, 28–29; Ex. 1009 ¶ 60).²³ On cited page 28, Sanders urges the user to “repeat this step to create an export entry for each clone volume that is used for the clone database.” Ex. 1013, 28. As quoted in the Institution Decision, Edwards specifically refers to the clone as a database accessible by applications: “[t]he clone database is a frozen image of the database file system at the

²³ In its Reply, Petitioner also points to Dr. Shenoy’s acknowledgement that the NetApp server can access NetApp disk blocks indirectly. *See* Pet. Reply 18–19 (citing Ex. 1031, 305:8–15 (“So as I just said, when a database block is read by a database server, you map it to some portion of a database file. You make a request to that portion of the data from the file system, which then maps it to underlying disk block.”)).

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.” Dec. on Inst. 16 (quoting Ex. 1013, 6).

Based on the foregoing discussion and consideration of the record, the record supports Petitioner’s showing.

c. Virtualization

In addition to the argument asserting the lack of virtual database in the NetApp system listed above (PO Resp. 39), Patent Owner also argues, as follows, that Edwards does not describe a virtual database:

This “virtualization” *is not the process of decoupling data from its storage layer.* (Shenoy Decl. at ¶ 113.) Rather, it is virtualization within the file system storage layer itself, *permitting a decoupling of file system structures from the physical disks which store data* and allowing instantiation of many file systems on a set of physical drives rather than confining a file system per a physical drive or dedicated set of disks.

Id. at 40.

According further to Patent Owner, “[o]ne skilled in the art would not, after reading Sanders, Edwards and Patel, know how to put together a system where a virtual database can run on a different database server and operating system than the production database to ensure, for example, the described flexibility in provisioning a test system.” PO Resp. 35 (citing Ex. 2017 ¶ 119). Patent Owner makes additional arguments advocating its view of “virtualization” that track Patent Owner’s claim construction arguments that we address above. *See supra* Section II.C; ’026 PO Resp. 35–36, 40, 55 (“Petitioner attempts to . . . draw a false equivalence between Flex volume ‘virtualization’ . . . and the claimed virtualization of data bases. The Flex

volumes operated at the physical storage level, not the application level, which is the level at issue in the challenged claims.”).

Again, Patent Owner’s arguments largely turn on claim construction and fail to rebut Petitioner’s showing. Patent Owner’s arguments regarding virtualization turn in part on the claim construction of database blocks and databases. *See* PO Resp. 39. These arguments also are not persuasive for the reasons explained above. As determined above, the NetApp system as described in Sanders and Edwards, and as further described in Data ONTAP Guide and Patel, is virtual and operates on database blocks according to the claim construction set forth above. In addition, contrary to Patent Owner’s arguments, the challenged claims do not require the ability to operate a virtual database with a different database server and operating system than the production base. They also do not require operation at the application level. As we determine above in several instances, the ’077 patent describes, and the claims embrace, file system databases. *See supra* Section II.C.1.i, ii & 2; *see, e.g.*, Ex. 1001, 7:4–10 (“FIG. 1 illustrates one embodiment for how information may be *copied* from a production database to a database storage system . . . *using a file sharing* system. . . . In some embodiments information may be *copied* from *storage level snapshots* of production databases.”) (emphases added).

Further regarding virtualization, Patent Owner’s arguments concede that the NetApp system as disclosed by Sanders employs a type of virtualization—“*a decoupling of file system structures from the physical disks which store data.*” PO Resp. 40 (emphasis added). Patent Owner’s characterization of “virtualization” changes in various arguments. For

example, Patent Owner states that with respect to the claimed invention, “[i]n reality, however, each virtual database is ‘virtual’ as it is accessing the same remote system and the same database blocks received at different points in time as other virtual databases.” *Id.* at 31. On the other hand, Patent Owner also states that the “‘virtualization’ [described in Edwards] is not the process of decoupling data from its storage layer.” *Id.* at 40 (citing Ex. 2017 ¶ 113).

As found and generally described above in connection with claim 1, the NetApp system described by Edwards and Sanders allows the creation of multiple virtual database clones, each of which may access and share the same database blocks received from a source databases at a destination database at different points in time via SnapMirror and FlexClone. For example, Petitioner persuasively relies on teachings in Sanders and Edwards that describe making clones that share blocks with a base snapshot and making writable copies for “development or test purposes.” *See* Pet. 35–36 (citing Ex. 1009 ¶ 135; Ex. 1014, 14–15). Petitioner also argues persuasively that “the physical implementation of the files in the clone volume is decoupled from the logical use of the files. The cloned volume is ‘virtual’ because it ‘points’ to the pre-existing data blocks associated with a point-in-time copy of its parent volume instead of physically duplicating the data blocks of its parent.” Pet. 31 (citing Ex. 1009, 122–124). As determined above, a “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.” *Supra* section II.C.2.iii (Claim Construction).

Patent Owner does not challenge Petitioner's contentions persuasively. Rather, apart from the "set of files" limitation discussed below, Patent Owner effectively concedes that the NetApp system described by Sanders and Edwards includes a virtual database according to our claim construction. Furthermore, the '077 patent Specification touts eliminating "[s]everal steps requiring *copies of [a] database . . .* as a result of the use of virtual databases." Ex. 1001, 25:61–63.

In other words, as Petitioner contends:

The purpose of the patent is virtualizing data by eliminating redundant physical copies, not platform independence. '077 patent at 20:66–21:2 ("workflow scenarios associated with databases can be simplified and made highly efficient by creating virtual databases instead of making physical copies of the databases."); *id.* at 1:64–67. The patent's database virtualization thus refers to virtualizing data. Mapping VDB files to the physical addresses of stored database blocks, rather than making physical copies, achieves the stated purpose.

Pet. Reply 16 (citing Ex. 1001, 19:53–56, 20–39–45, Figs. 10–11; Ex. 1063 ¶ 37); *see also* Pet. Reply 14 (citing Ex. 1031, 54:20–55:1 (Dr. Shenoy testifying that pointing to the "same structures" via a "new root inode" and the original root inode constitutes "in that sense . . . a virtual copy)); 15 (citing Ex. 1063 ¶ 37 and describing data virtualization); Ex. 1063 ¶¶ 36–44 (describing a virtual database as data virtualization in light of the '077 patent).

Describing the NetApp WAFL system as described in Edwards, Dr. Shenoy also states, during cross-examination, that "you made a copy of the root inode, which is pointing to the same structure. That's a virtual copy. It does involve pointers." Ex. 1031, 56:5–7. He also testifies that "[b]ecause

you do not have two physical copies, it's a virtual copy.” *Id.* at 55:4–5.²⁴ In a similar fashion, the '077 patent repeatedly highlights these essential features of a virtual database: “A point-in-time copy retrieves data changed in the database *A virtual database (VDB) is created by creating a set of files in the data storage system. Each file in the set of files created for a VDB is linked to the database blocks . . . associated with a point-in-time copy of the source database.*” Ex. 1001, Abstract (emphasis added); *see also id.* at 15:2–6 (noting that “[s]ubsequent PIT copies obtain only the changed information . . . and can be much smaller than the information contained in the first PIT copy.”)

Patent Owner also argues that our construction improperly covers a source database. PO Resp. 32. This argument, like most of Patent Owner's arguments, are addressed above in the Claim Construction section. *See supra* Section II.C.2.ii. Based in part on Patent Owner's arguments, as addressed, the modified claim construction clarifies that a virtual database requires association with another database: “a set of readable and writable database files capable of being mapped to physical addresses for stored

²⁴ Dr. Shenoy also contends that he is “not sure that there is any decoupling going on at that level” in a discussion of the WAFL system, where “the pointers are shared, the disk blocks are shared.” Ex. 1031, 54:17–19. Essentially Dr. Shenoy appears to contend that sharing pointers by “simply creating a copy of the root inode” makes him “not sure” of “any decoupling.” Ex. 1031, 54:15–551. Dr. Shenoy's testimony about “not [being] sure” fails to rebut sufficiently the record evidence (including his statements) about the meaning of virtual in light of the '077 patent Specification. Furthermore, Dr. Shenoy fails to explain why having a copy of an inode that points to another inode structure is any more decoupled from the underlying physical data than copying the whole inode structure. *See* the discussion below regarding the claimed “set of files.”

database blocks associated with another database.” *Supra* Section II.C.2.iii. Relying on pointers to the source database (i.e., another database), Petitioner shows that the NetApp system, as taught by Sanders and Edwards, renders this feature of virtualization to be part and parcel of, and rendered obvious by, the NetApp system. *See, e.g.*, Pet. 37–38 (explaining that creating a new vol_info block points to a PIT copy of “already-stored database blocks associated with a point-in-time copy of the source data base”); Ex. 1009 ¶¶ 140–42 (testimony directed to the same point).

As Petitioner summarizes: “The claims require ‘receiving database blocks for the point-in-time copies of the source database’ and creating files ‘linked to the stored database blocks.’ . . . A source database is not created by mapping or pointing to stored blocks of point in time copies of itself.” Pet Reply 15 (citing Ex. 1063 ¶ 36). As Petitioner also points out, and as discussed above in the claim construction section, “the patent states that a source database can be a virtual database.” *Id.* (quoting Ex. 1001, 22:14–16 (“The virtual database may be created based on . . . a point-in-time copy of another virtual database.”); *see supra* section II.C.2. In other words, a central essence of a virtual database according to the ’077 patent Specification involves eliminating the requirement to make physical copies: “Various workflow scenarios associated with databases can be simplified and made highly efficient *by creating virtual databases instead of making physical copies of the data bases.*” Ex. 1001, 20:66–21:2 (emphasis added).

Based on the foregoing discussion and consideration of the record, the record supports Petitioner’s showing.

d. Set of Files

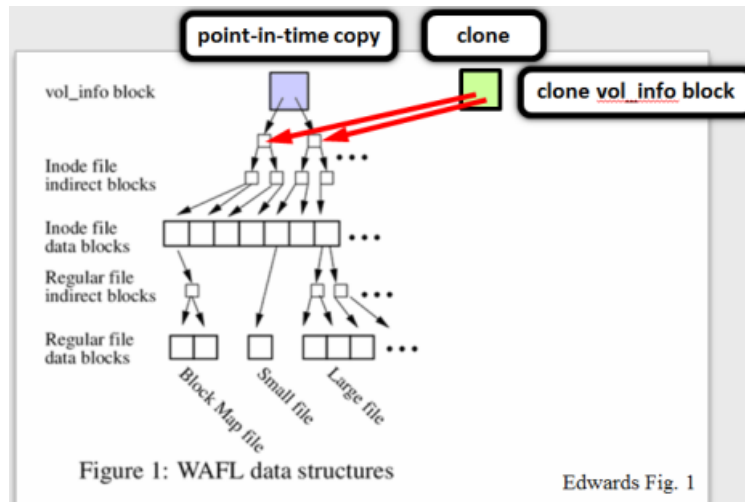
Patent Owner contends that creating a Snapshot in Edwards by creating “the new vol_info block is not a set of files, which is what must be linked to the stored database blocks. . . . Nor does the new vol_info block point to stored database blocks.” See PO Resp. 59 (citing Ex. 2017 ¶¶ 155–56). Patent Owner makes materially the same arguments in the ’026 IPR. See ’026 PO Resp. 56–59.

These arguments are not persuasive. The argument about stored “database blocks” again turns on Patent Owner’s unsupported claim construction of that term. The NetApp system discloses “database blocks” according to our claim construction. Patent Owner does not dispute, or rebut, persuasively, Petitioner’s contention that the NetApp system points to the NetApp data blocks.²⁵

As to the other argument, Patent Owner contends that each clone merely creates a “clone vol_info block,” which “is not a set of files.” *Id.* at 59 (emphasis added). Patent Owner explains that according to a dictionary source, an applicable meaning of “**set**” is “[a] group of things of the same kind that belong together and are so used.” See *id.* (citing Ex. 2027, 3–4). Petitioner disagrees with Patent Owner. See Pet. 36–39; Pet. Reply 22 (“The creation of a new vol_info block creates a new file system with new files.”) (citing Ex. 1009 ¶¶ 74–76; 1063 ¶¶ 55–57).

²⁵ For example, although Patent Owner contends that “[i]node files or inodes are not mapped to physical addresses for *database blocks*”; Patent Owner also states that “inode files are collections of inodes which may point to file system *data blocks*.” PO Resp. 54 (emphases added).

To illustrate the argument, Patent Owner annotates and reproduces the following version of Edward's Figure 1:



PO Resp. 59 (citing Ex. 1014, Fig. 1). As described by Patent Owner, above, annotated Figure 1 of Edwards illustrates Edwards's pointer system, which ultimately uses inode files to map to "data blocks." See PO Resp. 54–55, 58–59.

Contrary to Patent Owner's arguments, by creating a new "clone vol_info block," the NetApp system creates a new set of files. In other words, creating a new member, "clone vol_info block," which points to the old set of files pointing to PIT copies, constitutes creating a new "group of things of the same kind that belong together and are so used," according to Patent Owner's proffered definition of "set." See PO Resp. 59 (citing Ex. 2007, 3–4). By analogy, the set of numbers (1, 5, 6) is a different set than the set of numbers (2, 1, 5, 6), even though the sets overlap. Patent Owner's arguments imply that all original pointers in Edwards would have to be re-copied to constitute "creating a set of files" as set forth in the challenged claims, but Patent Owner fails to explain why that creates a patentable

distinction over the set of files as Petitioner advances with respect to Sanders and Edwards, and as described above. *See* PO Resp. 59; *supra* note 24; Ex. 1063 ¶ 55 (addressing the argument).

In addition, as Petitioner argues, FlexClone creates a new FlexVol volume: “[A] FlexVol volume is a file system created *within* a file on an underlying file system.” Pet. Reply 22 (quoting Ex. 1014, 11, citing Ex. 1013, 14; Ex. 1031, 257:4–6 (“[A] FlexVol is based on the WAFL file system. So in this case, there is a file system associated with the volume.”)). The clone vol_info block represents a file structure that corresponds to the “‘read/write file structure representing a VDB’ created in the patent.” *See* Pet. Reply 22 (quoting Ex. 1001, 19:16). In other words, as Petitioner argues, “the creation of a new vol_info block creates a new file system with new files.” *Id.*; *see also* Ex. 1009 ¶¶ 78–82 (summarizing Edwards as a virtual database system); Ex. 1063 ¶¶ 55–58. Furthermore, the NetApp system can create a clone of a clone quickly, thereby creating another set of files, at any point in time. *See* Ex. 1013, 3 (“a Flexclone volume has the ability . . . to be the source of another FlexClone volume”).

As described above in this section and the summary above of Edwards, the record supports Petitioner. Edwards discloses that

[c]reating a clone volume is a simple process. WAFL *creates the files required for a new FlexVol Volume. But rather than creating and writing a new file structure inside the volume, WAFL seeds the container file of the clone with a vol_info block that is a copy of the vol_info block of the Snapshot copy on which the clone is based.*
Ex. 1014, 15 (emphases added).

In other words, according to Edwards, WAFL “creates the files” (i.e., a new file system) during cloning, as Petitioner and Dr. Zadok contend, even if it does not write “a new file structure inside the volume.” *See id.* (noting other differences in the files of the created FlexClone volume, such as “holes” and “the writing of a fixed, small number of blocks”).

Expounding on the point, Dr. Zadok testifies that FlexClone volumes are writable and therefore different than snapshot’s read-only files, thereby showing for another reason that the clone vol_info block constitutes an “independent set of files.” Ex. 1063 ¶ 58, ¶¶ 55–58 (responding to Patent Owner and Dr. Shenoy). The record supports Dr. Zadok’s testimony in this regard also. *See* Ex. 1014, 15 (discussing FlexClone volumes); Ex. 1013, 3 (FlexClone volumes are “writable point-in-time image of a FlexVol volume or another FlexClone volume). As Dr. Zadok testifies, Dr. Shenoy agrees that a FlexClone volume (created with a cloned vol_info block) constitutes a writable file structure and a snapshot copy is read only. *See* Ex. 1063 ¶¶ 55–58; Ex. 1031:267:2–269:6; 351:19–352:8. Accordingly, as Petitioner argues, “the files of a FlexClone cannot be the same as the files of a snapshot.” Pet. Reply 23 (citing Ex. 1063 ¶ 58).

Based on the foregoing discussion and consideration of the record, the record supports Petitioner’s showing.

e. PIT Copies

Claim 1 requires “receiving database blocks for the point-in-time copies of the source database . . . storing the database blocks on a storage system; . . . [and] creating a set of files linked to the stored database blocks on the storage system.” Patent Owner also argues that “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 51. According to Patent Owner,

. . . when a copy of a FlexVol volume is created by SnapMirror, it is the exact replica of the current state of a FlexVol volume with all the inodes and pointers to data blocks is created. (Shenoy Decl. at ¶¶ 141, 148, 152.) Even if the FlexVol volume included various snapshots, those snapshots are merely copied when the entirety of FlexVol volume is replicated—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 ¶¶ 149–50.) And even if the volume includes snapshots, merely cloning the current state of the volume is fundamentally different than the invention of the ’077 patent, which allows receiving point-in-time copies of the source database at different points in time and creating a virtual database using the point-in-time copies. (*Id.* at ¶¶ 149–52.)

PO Resp. 51.

Patent Owner’s arguments, focusing on its disclosed invention as being “fundamentally different,” do not rebut Petitioner’s showing as summarized above with respect to a specific element as recited in challenged claim 1. Petitioner shows that Sanders discloses loading PIT copies during initial steps employed to clone a FlexVol using SnapMirror. *See* Pet. 33 (citing Ex. 1013, 21, 23; Ex. 1009 ¶ 129); Ex. 1009 ¶¶ 62 (citing Ex. 1013, Figs. 3, 4) 64 (citing Ex. 1013, 27, 28). Petitioner also replies that “the claims require ‘receiving a plurality of point-in-time copies.’ ’077 patent, claim 1. But these different PIT copies need not be *received* ‘at different points in time.’” Pet. Reply 20 (quoting PO Resp. 51).

Claim 1 requires the “storage system” to receive and store “database blocks for the [PIT] copies.” According to claim 1, the PIT copies (snapshots) each may be received and stored at different times, or at the same time, and the set of files are linked to them. Patent Owner’s argument admits that the NetApp system FlexVol volume includes a current state that “includes snapshots” (and that those snapshots are cloned).

As summarized above with respect to the Petition, creating NetApp snapshots at a destination satisfies the claim 1 requirements of receiving and storing database blocks for the PIT copies on a storage system. *See, e.g.*, Ex. 1013, 3 (stating “[d]ata changes made at SnapMirror source are continuously replicated at the SnapMirror destination”); 21 (during cloning, “the initialization process transfers data, including all Snapshot copies, from the source volume to the destination volume for the first time; thereafter, only changed blocks are transferred”). Therefore, as Petitioner contends, even if the claims require “receiving point-in-time copies of the source database *at different points in time*,” “SnapMirror sends snapshots stored on the source to the destination.” Pet Reply 20 (quoting PO Resp. 51 (emphasis by Petitioner), citing Ex. 1031, 107:2–108:2).

There is no dispute that a SnapMirror destination receives “continuously replicated” changes at different points in time. *See* Ex. 1013, 3. Edwards provides background to Sanders regarding PIT copies: “WAFL write anywhere [file layout scheme] . . . can create point-in-time Snapshot copies of a volume almost for free.” Ex. 1014, 11. Each time NetApp writes a block, it updates links or pointers to the new location on the block. Snapshot copies point to the blocks. *See* Ex. 1009 ¶¶ 71–77 (citing Ex.

1014, 11); Ex. 1014, 15 (“Like the creation of a snapshot, creating a flexible volume clone, or FlexClone volume, requires the writing of a fixed, small number of blocks and is nearly instantaneous.”).²⁶

Based on the foregoing discussion, the record supports Petitioner’s showing, and Patent Owner does not rebut it persuasively.

f. Linking, Volume, Database

Regarding the linking element in challenged claim 1, Patent Owner contends that the proposed combination does not disclose or render obvious identifying a “source database,” but rather, only identifies “a NetApp volume.” PO Resp. 41–42. Patent Owner contends that “an identification of a volume is not an identification of a database,” because “there is no necessary correspondence between” the two. *Id.* at 42 (citing Ex. 2017 ¶¶ 159–62. According to Patent Owner, “Petitioner’s expert acknowledged that multiple database[s] could be stored in a single volume.” *Id.* (citing Ex. 2016, 188:22–189:10)). Patent Owner also argues that Sanders does not disclose the “linking” step, because identifying the source database and linking constitute separate requirements. PO Resp. 43.

Patent Owner’s arguments are not persuasive. Regarding the latter argument, claim 1 recites “linking a source database, wherein linking the source database comprises receiving information identifying the source database.” As Dr. Zadok testifies, “[t]he [S]pecification explains that the required information is necessary ‘to retrieve data from the production database system 110.’” Ex. 1063 ¶ 52 (citing Ex. 1001, 21:10–18). The

²⁶ As discussed above, FlexClones are essentially writable snapshots. Ex. 1013, 3. Patel’s Figure 4 reproduced below presents a good illustration representing the copying of changed database blocks for a clone.

Specification states that “[t]he information may include the name of the database” and other information. It also states, “as part of the linking operation, the database storage system may communicate with the production database production system.” Ex. 1001, 21: 17–19. The ’077 Specification states further that the information “enables the database storage system 100 to retrieve data,” and “[t]he database storage system 100 can retrieve database blocks from the linked database,” but “[t]he load operation retrieves data” only from a “linked . . . database storage system.” Ex. 1001, 21: 15–17, 23–25, 35–39.

In other words, the ’077 patent Specification tracks claim 1, and describes providing some type of necessary information for linking, such as the name of the database, but does not describe any other necessary information or steps as part of linking. Moreover, Petitioner contends that “the ‘snapmirror initialize’ command is executed on the destination storage system and provided with the source volume information.” Pet. 32 (citing Ex. 1013, 22–23; Ex. 1009 ¶ 127). Therefore, Petitioner shows that this command links the destination and source databases together. Beyond arguing that the NetApp system does not disclose or suggest linking by merely providing a database name, Patent Owner fails to identify what additional action the term requires and fails to distinguish the initialize command persuasively. See PO Resp. 42–43.

Turning to the former arguments, as Petitioner and Dr. Zadok also contend, identifying a volume identifies a database. See Pet. Reply 20–21 (quoting Ex. 1013, 6 (“dbdata is “used to store the production database table data”); citing Ex. 1063 ¶ 52). Sanders describes “creat[ing] an instantaneous

point-in-time copy of a production database by creating an exact copy of the storage system volumes that are used to hold the database's data and transaction logs." Ex. 1013, 1. Sanders refers to "cloned databases" that "share data with the production database they were created from." *Id.* Sanders also teaches that "you need to initialize the SnapMirror relationship for each volume that is used for the dataset by executing" the initialize command. *Id.* at 22. Sanders also describes "[c]reating a DB2 Database on a NetApp or IBM N-Series Storage System" (*id.* at 9), refers to "the FlexVols used for the source database" (*id.* at 9, Fig. 4, step 2), refers to "[c]lone database volumes (*id.* at 8, Fig. 3), and uses a command "create database [DatabaseName] on [MountPoint]. As Dr. Zadok also testifies, "the SnapMirror initialize command identifies a source volume named 'dbdata' on a source storage system "srcstore." Ex. 1063 ¶ 53 (citing Ex. 1005, 22 ("snapmirror initialize -S srcstore:dbdata dststore:dbdata.")). Based on these teachings, an ordinarily skilled artisan would have recognized that a volume designated for and containing a database identifies a database. See Ex. 1063 ¶ 53.

Therefore, even if it is possible for a volume to contain more than one database as Patent Owner contends (see PO Resp. 42–43), Sanders at least discloses or suggests that a volume also includes a single database—in typical situations involving the cloning of a desired database. Petitioner shows that Sanders discloses, or at least suggests, linking by identifying a source database as set forth in claim 1, in order to clone a database.²⁷

²⁷ In Dr. Zadok's deposition testimony relied upon by Patent Owner (Ex. 2016, 188:22–189:10, PO Resp. 42), Patent Owner asks if "it is possible to

Based on the foregoing discussion and the record, Patent Owner does not rebut Petitioner's showing persuasively.

g. Claims 1, 3, 5–7, 9, 10, 18, 20–23, 25, and 26—Summary

We adopt Petitioner's showing as summarized above, and also with respect to claims 3, 5–7, 9, 10, 18, 20–23, 25, and 26. *See* Pet. 30–53. The Scheduling Order cautions Patent Owner that any arguments for patentability not raised in the Response will be deemed waived. *See* Paper 12, 3. As noted above, Patent Owner has grouped together independent claims 1 and 18 and focused arguments on claim 1. PO Resp. 41–42. Patent Owner has elected not to respond separately, with specificity, to the grounds of unpatentability asserted against claims 3, 5–7, 9, 10, 18, 20–23, 25, and 26. Based on the foregoing discussion and a review of the record, we determine that Petitioner has established by a preponderance of the evidence that the subject matter of claims 1, 3, 5–7, 9, 10, 18, 20–23, 25, and 26 would have been obvious over Sanders and Edwards.

v) Remaining Asserted Grounds

As noted above, the Scheduling Order cautioned Patent Owner that any arguments for patentability not raised in the Response will be deemed waived. *See* Paper 12, 3. As also noted above, Patent Owner has grouped

store more than one database in a volume,” and Dr. Zadok replies that “I believe the technology is flexible enough it allows you to store multiple sets of files or multiple databases within the same volume.” Dr. Zadok otherwise testifies that “the volume name, in my opinion, identifies this source database.” *Id.* at 188:19–20. The possibility raised by Patent Owner does not overcome Petitioner's showing of obviousness where NetApp discloses, or at least suggests, identifying a database via a volume name or vice versa.

together independent claims 1, 11, 18, and 27 and focused arguments on claim 1. PO Resp. 41–42; '026 PO Resp. 40–41. As summarized further below, Patent Owner has elected not to respond separately, with specificity, to the grounds of unpatentability asserted against claims 2, 4, 8, 11–13, 17, 19, 24, 27–29, and 33.

a. Claims 2, 4, and 19—Sanders, Edwards, and Patel

Petitioner adds the teachings of Patel to the basic combination of Sanders and Edwards in an asserted ground of obviousness as to dependent claims 2, 4, and 19. Pet. 52–57. Petitioner relies on Patel, a patent directed to NetApp, to demonstrate the obviousness of these claims. Pet. 52–57 (supported by the testimony of Dr. Zadok in Ex. 1009). These claims require associating a test virtual database with a PIT copy of a development database, or a clone of a clone. As noted above, and as Petitioner contends, Sanders teaches cloning a clone (i.e., using a clone as a source database), and Patel teaches using a test clone as a source to clone a quality assurance clone. See Pet. 54–55 (citing Ex. 1009 ¶¶ 150, 157; Ex. 1013, 3); Pet. 25–26 (citing Ex. 1016, 1, 3, 10, Fig. 10; Ex. 1009 ¶¶ 83, 84).

In addition, as found above, Patel describes the NetApp technologies described by Edwards and Sanders (i.e., FlexClone, SnapMirror, and Snapshot) as “solutions [that] dramatically reduce the effort required to deploy databases for application development, testing, QA, maintenance, and migration.” See Ex. 1016, 1, 6, 7. According to Patel, “NetApp Snapshot technology offers advantages because a Snapshot copy does not involve copying data from one location to another, so it is extremely fast and space efficient.” *Id.* at 8.

Patel's Figure 4 follows:

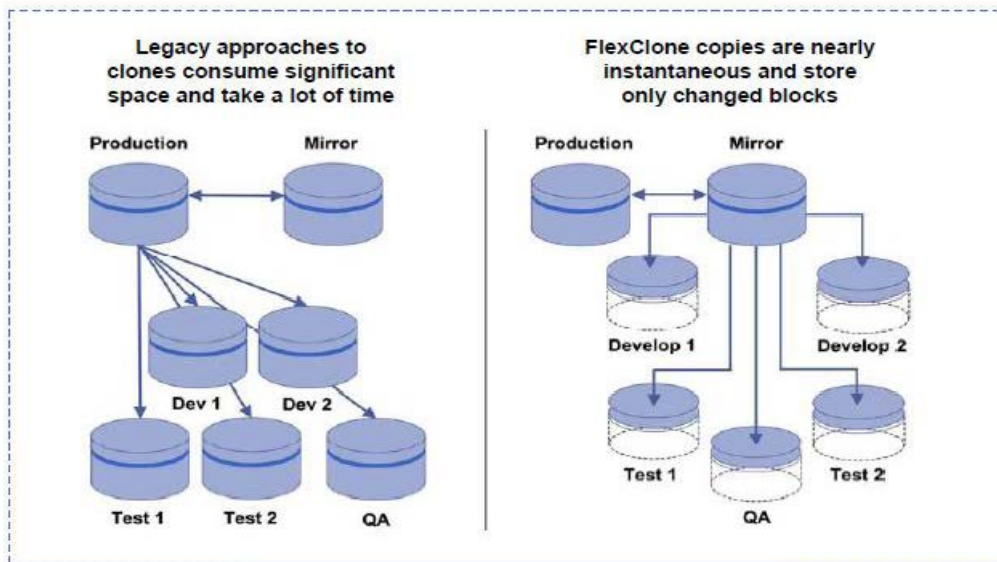
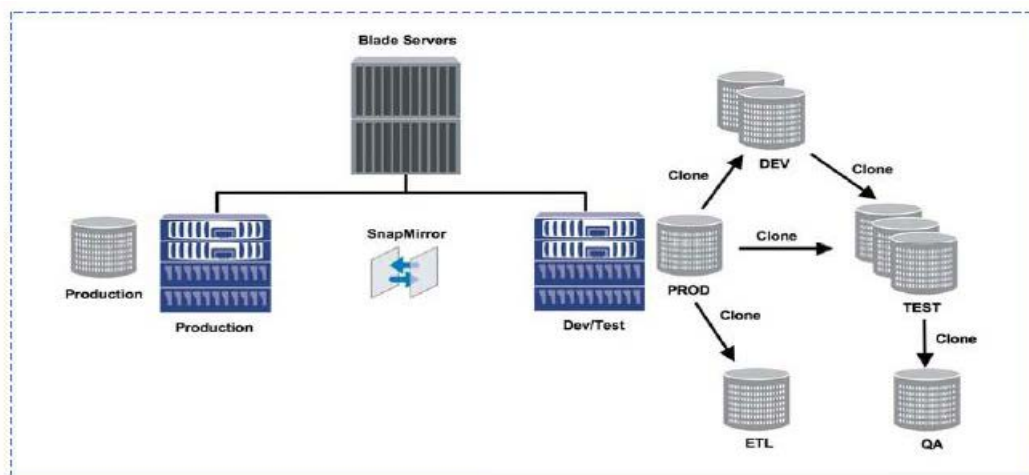


Figure 4) FlexClone enable the instant creation of multiple database clones that require only minimal additional storage.

Ex. 1016, 8. In general, Figure 4 represents the creation of multiple FlexClone copies (e.g., QA, Test 1), which store only changed blocks. According to Patel, these copies can be created and provisioned faster and with less memory space than legacy systems to provide development, test, and QA databases. *See id.* at 8.

Patel's Figure 2 follows:



Patel's Figure 2 represents using NetApp's SnapMirror to replicate a production database, and provisions test and QA clones from a development clone. *See id.* at 6.

Patent Owner directs arguments to claim 1 and generally attacks Petitioner's showing regarding motivation or reasons to combine Patel with the base references, as discussed above. These arguments are not persuasive for the reasons discussed above. We adopt Petitioner's showing and determine that Petitioner has established by a preponderance of the evidence that the subject matter of claims 2, 4, and 19 would have been obvious over Sanders, Edwards, and Patel.

b. Claims 8, 11–13, 24, and 27–29—Sanders, Edwards, and Data ONTAP Guide

Petitioner adds the teachings of Data ONTAP Guide to the basic combination of Sanders and Edwards in an asserted ground of obviousness as to dependent claims 8, 11–13, 24, and 27–29. '026 Pet. 25–33, 43–57. Petitioner uses the teachings of Data ONTAP Guide to provide further details of NetApp's storage copy features to show the obviousness of dependent claims 8, 11–13, 24, and 27–29, again relying on the testimony of Dr. Zadok. *Id.* Dr. Zadok testifies that Sanders expressly refers to Data ONTAP Guide as a technical reference. '026 Ex. 1109 ¶ 109 (citing '26 Ex. 1113, 50 (i.e., same as Exhibit 1013)). Petitioner primarily relies on Data ONTAP Guide's volume copy feature, to address the staging limitations of the challenged dependent claims, and to bolster the staging teachings in Sanders. *See, e.g.*, '026 Pet. 43–45 (citing '026 Ex. 1113, 5; '026 Ex. 1009 ¶¶ 166–67, '026 Ex. 1117, 379, 381). Petitioner contends that Data ONTAP

Guide's NetApp teachings would have been useful to supplement Sanders's NetApp teachings and provide volume copies in order to provide disaster protection. *Id.* at 44–45; *see also id.* at 29–30 (discussing further motivation to combine).

As indicated above, Patent Owner directs arguments to claim 1, and generally attacks Petitioner's showing regarding motivation or reasons to combine Data ONTAP Guide with the base references, as discussed above. *See also* '026 PO Resp. 34–35 (repeating general arguments applicable to claim construction and claim regarding database blocks, application level solutions, and metadata). Patent Owner does not rebut Petitioner's showing persuasively. We adopt Petitioner's showing and determine that Petitioner has established by a preponderance of the evidence that the subject matter of claims 8, 11–13, 24, and 27–29 would have been obvious over Sanders, Edwards, and Data ONTAP Guide.

c. Claims 17 and 33—Sanders, Edwards, Data ONTAP Guide, and Sarma

Petitioner adds the teachings of Sarma to the basic combination of Sanders, Edwards, and Data ONTAP Guide in an asserted ground of obviousness as to dependent claims 17 and 33. '026 Pet. 57–59. Petitioner uses the teachings of Sarma to supplement NetApp's storage system, as taught by the base references, to show the obviousness of dependent claims 17 and 23, again relying on the testimony of Dr. Zadok. *Id.* Patent Owner directs arguments to claim 1 and generally contends that Sarma does not relate to databases or virtualization, but Patent Owner does not address persuasively Petitioner's showing regarding the reasons for adding the teachings of Sarma to the base references. *See* '026 PO Resp. 40.

Petitioner relies on Sarma for its data compression and other techniques, not for its teachings about virtualization and databases. Pet. 58 (citing Ex. 1018, 2:27–11). Petitioner points out that Sarma discloses the NetApp SnapMirror and ONTAP technology. *Id.* (citing Ex. '026 Ex. 1018, 7:11–14; 14:16–23). Petitioner persuasively shows that Sarma's standard compression or encryption techniques would have benefited the NetApp system described in the base references, for example, "because compression was a well-known method of increasing transmission and storage efficiency." *Id.* (citing '026 Ex. 1108 ¶¶ 183–84); *see also id.* at '026 Pet. 31–33 (discussing further motivation to combine). We adopt Petitioner's showing and determine that Petitioner has established by a preponderance of the evidence that the subject matter of claims 17 and 33 would have been obvious over Sanders, Edwards, Data ONTAP Guide, and Sarma.

IV. CONCLUSION

Petitioner has met its burden of proof, showing by a preponderance of the evidence that claims 1–13, 17–29, and 33 of the '077 patent are unpatentable based on the following grounds under 35 U.S.C. § 103(a) for obviousness:

A. Claims 1, 3, 5–7, 9, 10, 18, 20–23, 25, and 26 over the combination of Edwards and Sanders;

B. Claims 2, 4, and 19 over the combination of Sanders, Edwards, and Patel;

C. Claims 8, 11–13, 24, and 27–29 over the combination of Sanders, Edwards, and Data ONTAP Guide; and

D. Claims 17 and 33 over the combination of Sanders, Edwards, Data ONTAP Guide, and Sarma.

V. ORDER

In consideration of the foregoing, it is hereby:

ORDERED that proceedings in IPR2015-00026 and IPR2015-00026 are consolidated for purposes of issuance of this final decision;

FURTHER ORDERED that claims 1–13, 17–29, and 33 of the '077 patent are *unpatentable*;

FURTHER ORDERED that Patent Owner's Motions to Exclude as to Exhibits 1029, 1030, 1032, 1033, 1035–1146, 1054, and 1062, and '026 Exhibits 1134–1136, 1138–1148, 1150, 1151, 1153, and 1156, are *dismissed* (as moot);

FURTHER ORDERED that Patent Owner's Motions to Exclude as to Exhibits 1048 and 1049, and '026 Exhibits 1117 and 1150, are *denied*;

FURTHER ORDERED that the parties' Demonstratives for the Oral Hearing (Papers 63–66, Exhibits 1067 and 1068, '026 Papers 60–63, and '026 Exhibits 1171 and 1172) are *expunged*;

FURTHER ORDERED that Petitioner's Motions to expunge Exhibit 1028 and '026 Exhibit 1130 are *granted*;

FURTHER ORDERED that the parties' Motions to Seal the Demonstratives (Papers 60 and 62, and '026 Papers 56 and 58) are *dismissed* (as moot); and

FURTHER ORDERED that Petitioner's Objections to Patent Owner's Demonstratives (Paper 67 and '026 Paper 64) are *dismissed* (as moot).

This is a Final Decision. Parties to the proceeding seeking judicial review of the decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

PETITIONER:

Robert Steinberg
Jonathan Link
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