

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SAP SE,
Petitioner,

v.

TERADATA US, INC.,
Patent Owner.

IPR2021-01017
Patent 7,904,419 B1

Before LYNNE E. PETTIGREW, BART A. GERSTENBLITH, and
RUSSELL E. CASS, *Administrative Patent Judges*.

PETTIGREW, *Administrative Patent Judge*.

DECISION
Denying Institution of *Inter Partes* Review
35 U.S.C. § 314

I. INTRODUCTION

Petitioner, SAP SE, filed a Petition for *inter partes* review of claims 1, 2, 4–12, 14, 15, and 17–28 of U.S. Patent No. 7,904,419 B1 (Ex. 1001, “the ’419 patent”). Paper 1 (“Pet.”). Patent Owner, Teradata US, Inc., filed a Preliminary Response. Paper 6 (“Prelim. Resp.”).

Under 35 U.S.C. § 314 and 37 C.F.R. § 42.4(a), we have authority to institute an *inter partes* review if “the information presented in the petition . . . and any response . . . shows that there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition.” 35 U.S.C. § 314(a) (2018). After considering the Petition, the Preliminary Response, and the evidence of record, we determine the information presented does not show a reasonable likelihood that Petitioner would prevail in establishing the unpatentability of at least one of the challenged claims of the ’419 patent. Accordingly, we do not institute an *inter partes* review.

II. BACKGROUND

A. Real Parties-in-Interest

Petitioner identifies SAP SE, SAP America, Inc., and SAP Labs, LLC as real parties-in-interest. Pet. 87. Patent Owner identifies Teradata US, Teradata Operations, Inc., and Teradata Corporation as real parties-in-interest. Paper 4, 1 (Patent Owner’s Mandatory Notices).

B. Related Matters

The parties identify the following district court proceeding involving the ’419 patent: *Teradata US, Inc. v. SAP SE*, Case No. 3:20-cv-06127 (N.D. Cal.), filed August 31, 2020. Pet. 87; Paper 4, 1.

C. Overview of the '419 Patent

The '419 patent is directed to managing allocation of temporary and permanent files in a database system. Ex. 1001, 1:7–8. As background, the '419 patent explains that typical database systems divide data into different storage regions including permanent storage, typically used to store user data, and temporary storage, typically used to store intermediate or final results of a query. *Id.* at 1:21–29. The '419 patent describes inefficiencies in prior database systems associated with “the separation of the database control logic for temporary and permanent files.” *Id.* at 1:48–50.

The '419 patent discloses a “file management context” to indicate allocation of files in storage. *Id.* at 1:56–64, 3:11–14. In one embodiment, a file management context “includes two maps: a first map to allocate storage identifiers for uniquely identifying each file created in the database system, and a second map to allocate allocation units, which are blocks or pages of storage.” *Id.* at 3:27–31. One embodiment of a database system includes a file management context in non-persistent memory to indicate allocation of temporary and permanent files and a file management context in a persistent storage device to indicate allocation of permanent files. *Id.* at 4:10–14.

Figure 2 of the '419 patent, reproduced below, illustrates an example of file management contexts stored in a database system (*id.* at 2:6–8):

FIG. 2

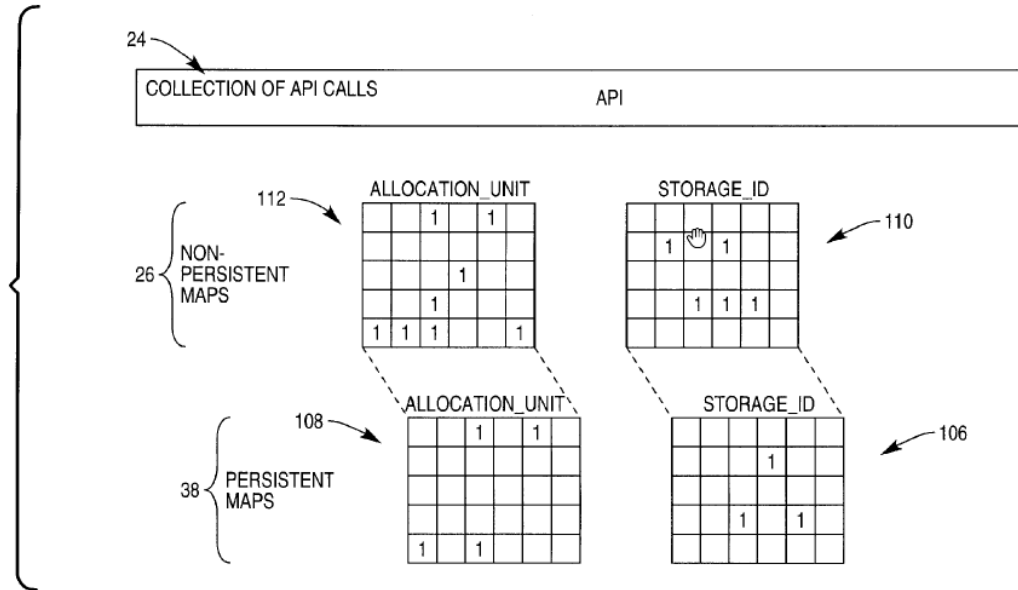


Figure 2 above shows file management context 26 in non-persistent memory and file management context 38 in a persistent storage device. *Id.* at 3:63–67. Persistent file management context 38 includes STORAGE_ID map 106 to allocate storage identifiers to permanent files and ALLOCATION_UNIT map 108 to allocate blocks or pages to permanent files. *Id.* at 4:1–4. Non-persistent file management context 26 includes STORAGE_ID map 110 to allocate storage identifiers to both temporary and permanent files and ALLOCATION_UNIT map 112 to allocate blocks or pages to both temporary and permanent files. *Id.* at 4:8–14. Each map includes an array of flags set to “1” or “0,” where “1” represents an active state indicating that a storage ID or allocation unit has been allocated. *Id.* at 4:4–7, 4:15–19. “[P]ersistent maps 106 and 108 are subsets of respective non-persistent maps 110 and 112.” *Id.* at 4:14–15.

In the event of a system crash or reset, maps 106 and 108 stored in a persistent storage device are maintained “so that permanent files are not lost as a result of the system crash or reset.” *Id.* at 5:60–64. By contrast, maps 110 and 112 stored in non-persistent memory are lost when the system crashes, “which effectively erases any allocated temporary files.” *Id.* at 5:64–65.

D. Illustrative Claim

Challenged claims 1, 2, and 4 are independent; claims 5–10, 18–24, 27, and 28 depend directly or indirectly from claim 1; claims 11, 12, and 25 depend directly or indirectly from claim 2; and claims 17 and 26 depend directly from claim 4. Claim 4, reproduced below, is illustrative for purposes of the issues raised at this stage of the proceeding:

4. An article comprising at least one storage medium containing instructions that when executed cause a system to:
 - store a first file management file context in non-persistent memory to indicate allocation of temporary and permanent files; and
 - store a second file management file context in persistent storage to indicate an allocation of permanent files.

Ex. 1001, 7:11–17.

E. Asserted Grounds of Unpatentability

Petitioner asserts that claims 1, 2, 4–12, 14, 15, and 17–28 are unpatentable based on the following grounds (Pet. 14):

Claim(s) Challenged	35 U.S.C. §	Reference(s)
1, 2, 4–8, 11, 14, 17–19, 21, 22, 28	103(a) ¹	System R, ² Lorie ³
9, 10, 12, 15, 20, 24–27	103(a)	System R, Lorie, Astrahan ⁴
23	103(a)	System R, Lorie, R* ⁵

III. DISCUSSION

A. 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

¹ The Leahy-Smith America Invents Act, Pub. L. No. 112-29, 125 Stat. 284 (2011) (“AIA”), amended 35 U.S.C. § 103. Because the ’419 patent has an effective filing date before the effective date of the applicable AIA amendments, we refer to the pre-AIA version of 35 U.S.C. § 103.

² Jim Gray et al., *The Recovery Manager of the System R Database Manager*, Computing Surveys, Vol. 13, No. 2, 223–42, June 1981 (Ex. 1002, “System R”).

³ Raymond A. Lorie, *Physical Integrity in a Large Segmented Database*, ACM Transactions on Database Systems, Vol. 2, No. 1, 91–104, Mar. 1977 (Ex. 1008, “Lorie”).

⁴ M. M. Astrahan et al., *System R: Relational Approach to Database Management*, ACM Transactions on Database Systems, Vol. 1, No. 2, 97–137, June 1976 (Ex. 1003, “Astrahan”).

⁵ Haas et al., *R*: A Research Project on Distributed Relational DBMS*, Quarterly Bulletin of the IEEE Computer Society Technical Committee on Database Engineering, Vol. 5, No. 4, 28–32, Dec. 1982 (Ex. 1009, “R*”).

obviousness is resolved on the basis of underlying factual determinations, including (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of ordinary skill in the art; and (4) when in evidence, objective indicia of non-obviousness.⁶ *Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966).

B. Level of Ordinary Skill in the Art

Citing the Declaration of Dr. Wolfgang Lehner, Petitioner contends that a person of ordinary skill in the art at the time of the invention “would have possessed at least a Bachelor’s degree in computer science with at least two years of experience with database system development.” Pet. 14 (citing Ex. 1006 ¶¶ 21–22). Petitioner further contends that a person of ordinary skill in the art could have had “some combination of (1) more formal education and less technical experience, or (2) less formal education and more technical or professional experience.” *Id.* (citing Ex. 1006 ¶¶ 21–22). Patent Owner does not dispute Petitioner’s proposed level of ordinary skill in the art. *See generally* Prelim. Resp.

At this stage of the proceeding, with the exception of the open-ended phrase “at least,” which expands the range of experience indefinitely without an upper bound, we adopt Petitioner’s assessment of the level of ordinary skill in the art because it is consistent with the ’419 patent and the asserted prior art. *See Okajima v. Bourdeau*, 261 F.3d 1350, 1355 (Fed. Cir. 2001).

⁶ With respect to the fourth *Graham* factor, the parties at this time do not present arguments or evidence regarding objective indicia of non-obviousness. Therefore, the obviousness analysis at this stage of the proceeding is based on the first three *Graham* factors.

C. Claim Construction

In this *inter partes* review, we apply the same claim construction standard that would be used in a civil action under 35 U.S.C. § 282(b). 37 C.F.R. § 42.100(b) (2019). In applying this standard, we generally give claim terms their ordinary and customary meaning as would be understood by a person of ordinary skill in the art at the time of the invention and in the context of the entire patent disclosure. *See id.*; *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312–14 (Fed. Cir. 2005) (en banc).

Petitioner contends that no claim terms require explicit construction at this stage of the proceeding. Pet. 16. Patent Owner contends that in related district court litigation, Petitioner has proposed construing several terms in accordance with pre-AIA 35 U.S.C. § 112 ¶ 6, and Patent Owner faults Petitioner for not applying those constructions in the Petition. Prelim. Resp. 7–8.

Based on the record before us, we need not construe any claim terms expressly to decide the issues presented at this stage of the proceeding. *See Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017) (holding that only claim terms in controversy need to be construed, and only to the extent necessary to resolve the controversy (citing *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999))).

D. Asserted Obviousness over System R and Lorie

Petitioner contends that claims 1, 2, 4–8, 11, 14, 17–19, 21, 22, and 28 of the ’419 patent are unpatentable under 35 U.S.C. § 103(a) as obvious over System R and Lorie. Pet. 24–70. In response, Patent Owner argues that Petitioner fails to show that the combination of System R and Lorie teaches a “file management context” stored in “non-persistent memory,” as required

by independent claims 1, 2, and 4. Prelim. Resp. 16–23. For the reasons discussed below, we determine that the information presented does not show a reasonable likelihood that Petitioner would prevail in establishing that claims 1, 2, 4–8, 11, 14, 17–19, 21, 22, and 28 would have been obvious over System R and Lorie.

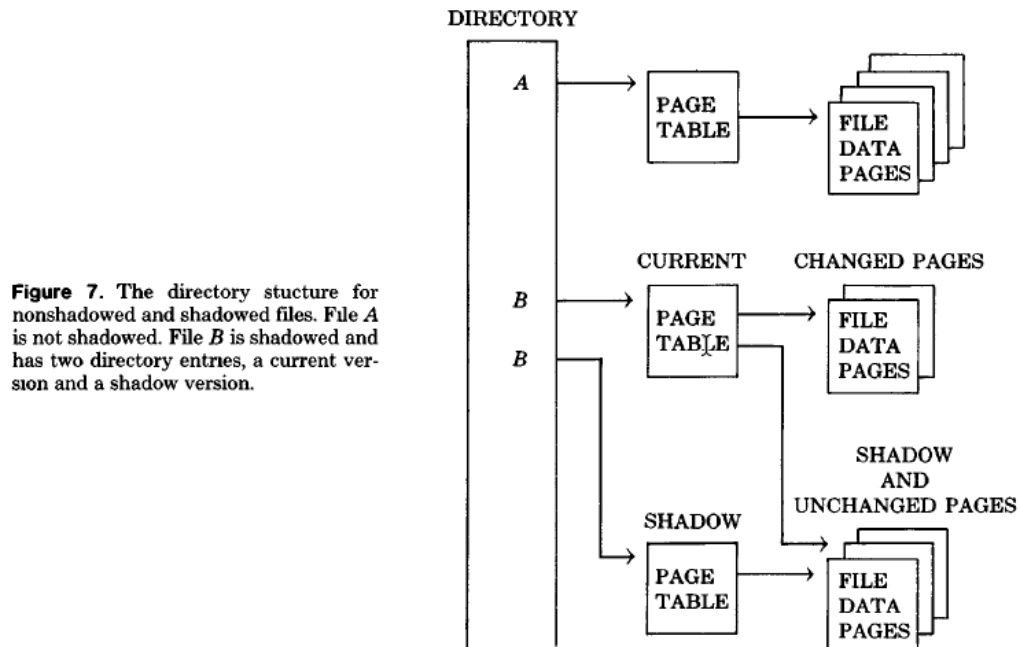
1. Overview of System R

System R describes the recovery manager of the Research Storage System (RSS) layer of the System R database system. Ex. 1002, 223 (Abstract), 224. The RSS supports the notion of a transaction, which consists of an application-specified sequence of RSS actions, including “operations to create, destroy, manipulate, retrieve, and modify RSS objects” such as files. *Id.* at 225, 227. An application declares the beginning of a transaction with a BEGIN action and the end of the transaction with a COMMIT action. *Id.* If the transaction either aborts or is aborted before the transaction commits, the system must undo all actions of that transaction. *Id.* at 227.

System R discloses that when the system crashes, the state of primary memory is lost but the state of disk storage is preserved. *Id.* at 226. When the system restarts following a crash, “the most recently committed state is reconstructed from the surviving disk state by referencing a log of recent activity to restore the work of committed and aborted transactions.” *Id.*

System R discloses that all persistent data are stored in files that are made up of 4096-byte pages allocated on disk. *Id.* at 229. A buffer manager brings files to be modified into a virtual memory buffer pool. *Id.* The buffer manager uses a least recently used (LRU) algorithm to regulate occupancy of pages in the buffer pool. *Id.*

The System R database stores two types of files, nonshadowed and shadowed, each having a particular recovery protocol. *Id.* at 230. Figure 7 of System R, reproduced below, shows the directory structure for shadowed and nonshadowed files:



Id. Figure 7 above shows a directory structure with a single entry for a nonshadowed file (file A) and two entries—a current version and a shadow version—for a shadowed file (file B). *Id.*

Nonshadowed files have no automatic recovery. *Id.* The system updates pages of nonshadowed files in the buffer pool and stores updated pages back to disk when the pages are removed from the buffer pool by the LRU algorithm or when the file containing the pages is saved or closed. *Id.*

“By contrast, the RSS maintains two on-line versions of shadowed files, a *shadow version* and a *current version*.” *Id.* “RSS actions affect only the current version of a file and never alter the shadow version,” unless the current version is saved as the new shadow version to make recent updates to the file permanent. *Id.* “Although the current version of a [shadowed]

file does not survive restart, because recent updates to the file may still reside in the buffer pool, the shadow version of a file does.” *Id.* “Hence at RSS restart . . . all shadowed files are reset to their shadow versions.” *Id.* System R refers to Lorie for a more detailed description of how the current and shadow versions of a shadowed file are implemented. *Id.* at 231.

2. Overview of Lorie

Lorie discloses a recovery scheme for use with the System R database. Ex. 1008, 91 (Abstract), 104. The recovery scheme “is based on maintaining a dual mapping between pages and their location on disk. One mapping represents the current state of a segment being modified; the other represents a previous backup state.” *Id.* at 91 (Abstract).

Lorie first describes a conventional technique for mapping logical pages of a database into physical slots of a segment on disk. *Id.* at 93–96. A mapping is implemented using a vector V_k , referred to as a page table, which is a sequence of slot numbers. *Id.* at 94. The i th element of the vector indicates which physical slot is used to store the contents of page i . *Id.* When pages need to be modified, the system brings the pages from their disk slots into a page buffer (PB) in memory for modification. *Id.* at 95. When processing is completed, the system writes pages back to disk if they have been modified. *Id.* Lorie explains that this conventional method is vulnerable to a system failure, such as a crash while the system is writing pages back to disk. *Id.* at 95–96. In that situation, “[i]f only some pages have not yet been written back on disk, the contents of these pages will be different from what the user expects.” *Id.* at 96.

Lorie proposes an improved recovery scheme that uses two versions of a logical page—a current and a backup. *Id.* at 96–97. Lorie discloses a current page table, V_{k0} , and a backup (or shadow) page table, V_{k1} , identifying

disk slots storing current and backup pages in disk segment S_k . *Id.* at 97, Fig. 3. To modify page i of S_k , the system retrieves page i from slot j on disk and loads it into the page buffer (PB) in memory, where it is modified. *Id.* When the system rewrites the modified page to disk, it writes the page into a new free slot j' instead of its original slot j . *Id.* The old slot j is kept as a shadow slot containing an older version of the page, and the new slot j' contains the current version of the page. *Id.* The current page table V_{k0} is updated to map the logical page to slot j' . *Id.* If the same page is modified again, it can be written back to the same slot j' . *Id.*

Figure 4, reproduced below, illustrates the database in a situation when disk segment S_l is open for modification:

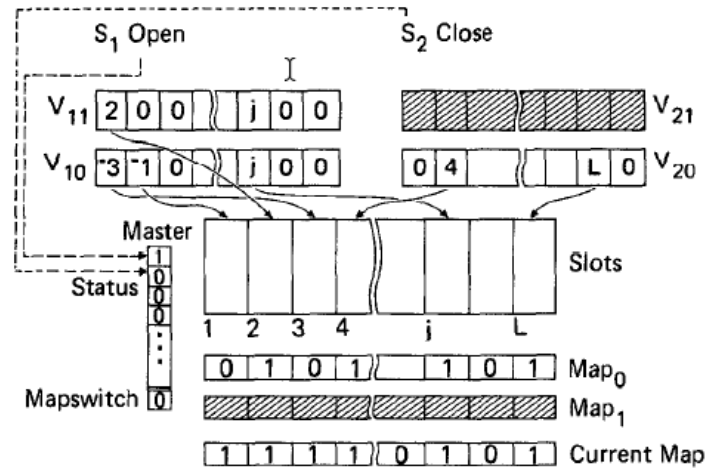


Fig. 4. Database representation— S_l modified

Id. at 98. Figure 4 above shows current page table V_{10} pointing to current versions of pages stored in slots on disk and shadow page table V_{11} pointing to shadow versions of pages stored in slots on disk. *Id.* If the database fails and then restarts, the system determines whether the segment was open (i.e., whether pages were being modified). *Id.* If the segment was open, shadow page table V_{11} reflects the last state of integrity and can be restored as the current page table. *Id.* Figure 4 also shows bit maps associated with the

page tables indicating whether slots of the disk segment are busy (denoted by a “1”) or free (denoted by a “0”). *See id.* at 95 (describing use of bit maps).

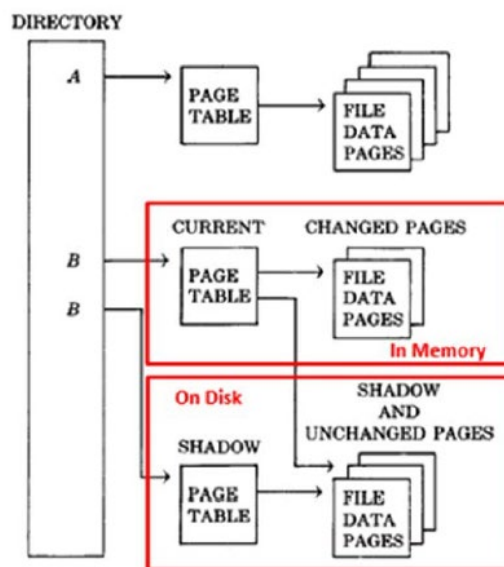
3. Analysis of Independent Claims 1, 2, and 4

Independent claims 1, 2, and 4 each require a “file management context” stored in “non-persistent memory” for indicating allocation of temporary and permanent files. Claim 4 recites an “article comprising at least one storage medium containing instructions . . . to . . . *store a first file management context in non-persistent memory to indicate allocation of temporary and permanent files.*” Ex. 1001, 7:10–14 (emphasis added). Claim 1, directed to a “database system,” likewise recites “*a non-persistent memory storing a second file management context . . . to indicate allocated temporary files and permanent files in [a] pool of storage elements.*” *Id.* at 6:43–52 (emphasis added). And claim 2, directed to a “method for use in a database system having a persistent storage device and a non-persistent memory,” similarly recites the step of “*storing a second file management context in the non-persistent memory,*” and requires “updating” the second file management context “*to allocate a permanent file*” and “*to allocate a temporary file.*” *Id.* at 6:53–63 (emphasis added).

Petitioner contends that System R in view of Lorie teaches or renders obvious a “file management context” stored in “non-persistent memory” for indicating allocation of temporary and permanent files, as recited in the independent claims. Pet. 28–35 (claim 4), 48–49 (claim 2), 52–58 (claim 1). In its analysis of claim 4, Petitioner argues that System R’s current page table and associated data structures, including free-space bit maps, comprise a “file management context” to indicate allocation of temporary and permanent files. *Id.* at 29–30 (citing Ex. 1002, 240), 35. Petitioner contends

that the functionality of System R's current page table and free-space bit maps, as further detailed in Lorie, mirrors that of non-persistent STORAGE_ID map 110 and non-persistent ALLOCATION_UNIT map 112 (i.e., non-persistent file management context 26) in the '419 patent. *Id.* at 35 (comparing Lorie's Figure 4 with Figure 2 of the '419 patent); *see also id.* at 5 (comparing Figure 7 of System R with Figure 2 of the '419 patent). Petitioner also contends that the System R current page table indicates "allocation of temporary [files] (Changed Pages containing records updated by yet-uncommitted transactions that do not survive a system restart) and permanent files (Unchanged/Shadow Pages stored on disk that do survive a system restart)." *Id.* at 35.

For the requirement that the file management context indicating allocation of temporary and permanent files be stored in "non-persistent memory," Petitioner refers to an annotated version of Figure 7 of System R, reproduced below:



Id. at 31; *see* Ex. 1002, 230. As shown above in red, Petitioner annotates Figure 7 to indicate that System R's current page table and changed pages

are “In Memory” and the shadow page table and shadow and unchanged pages are “On Disk.” Pet. 31; *see also id.* at 4, 20, 33, 36, 37, 39, 42, 43, 45, 46, 48, 49, 50, 51, 53, 54, 59, 60, 63, 64, 66, 67, 69, 77, 79, 81, 83 (duplicating Figure 7 with the same annotations throughout the Petition). Petitioner repeatedly describes the current page table as being “in memory.” *See, e.g., id.* at 5, 22, 46, 68.

Petitioner contends that a person of ordinary skill in the art “would have understood that the Current Page Table resides in non-persistent memory and tracks in-memory versions of Changed Pages (containing ‘temporary’ data records that do not survive a system restart) that have changed in the course of a current transaction.” *Id.* at 31 (citing Ex. 1002, 230). Petitioner further contends that a person of ordinary skill in the art “would have understood that the Current Page Table/Changed Pages reside in non-persistent memory because ‘the current version of a file does not survive restart.’” *Id.* at 31–32 (quoting Ex. 1002, 230). In contrast, Petitioner continues, “‘the shadow version of a file does’” survive restart. *Id.* at 32 (quoting Ex. 1002, 230). Petitioner also contends that “[w]hen performing RSS actions, System R operates on in-memory, and thus temporary, versions of pages,” and that “temporary augmentations of data records . . . are stored in Changed Page files in memory.” *Id.*

For the similar limitation in claim 2 (“storing a second file management context in the non-persistent memory”), Petitioner refers back to its analysis for claim 4. *Id.* at 48–49 (“And as described with reference to the first limitation of claim 4, System R’s Current Page Table and associated data structures such as free-space bit maps disclose a second file management context in non-persistent memory.”). Likewise, for claim 1’s recitation of “a non-persistent memory storing a second file management

context,” Petitioner refers back to its analysis for claim 2, which refers to the claim 4 analysis. *Id.* at 53 (“As described with reference to the second limitation of claim 2, System R’s Current Page Table and associated data structures such as free-space bit maps disclose a second file management context in non-persistent memory.”).

On the present record, we agree with Patent Owner that Petitioner does not show sufficiently that the combination of System R and Lorie teaches a “file management context” stored in “non-persistent memory.” *See* Prelim. Resp. 16–22. As an initial matter, Petitioner asserts that the “Current Page Table and associated data structures such as free-space bit maps” comprise the claimed “file management context” in “non-persistent memory,” but Petitioner does not cite any passage in System R disclosing that the current page table (or any associated data structure) resides “in memory” rather than “on disk,” as shown in Petitioner’s annotated version of System R’s Figure 7. *See* Pet. 28–35. Nor does Petitioner point to any disclosure in Lorie in support of its contention that the current page table in the System R database system is stored in non-persistent memory. *See id.*

Instead, Petitioner argues that a person of ordinary skill in the art would have understood that System R’s current page table resides in non-persistent memory because it tracks changed pages, which themselves “reside in non-persistent memory because ‘the current version of a file does not survive restart.’” *Id.* at 31–32 (quoting Ex. 1002, 230). On the record before us, the Petition lacks a sufficiently detailed explanation with evidentiary support for why a person of ordinary skill in the art would have concluded that the current page table shown in Figure 7 of System R resides in non-persistent memory. Notably, the Petition does not specifically cite the testimony of Dr. Lehner to support its contention regarding what a

person of ordinary skill in the art would have understood. *See id.* (citing System R but not Dr. Lehner’s Declaration). In any event, Dr. Lehner makes substantially the same statement in his Declaration, without any further explanation or citation to the evidentiary record. *See* Ex. 1006 ¶ 75.

Moreover, Petitioner’s contention that an ordinarily skilled artisan would have understood that System R’s current page table is present in memory rather than on disk is premised on Petitioner’s assertion that the changed pages in Figure 7 of System R, which the current page table tracks, reside in memory. Pet. 31–32. Patent Owner, however, identifies some apparent deficiencies in Petitioner’s position. Prelim. Resp. 19–21. For instance, Patent Owner argues that “Lorie makes it clear that both a current version and a shadow version of a logical page are stored in slots of a disk segment.” *Id.* at 19–20. As Patent Owner points out, Lorie’s Abstract “explicitly states that the disclosed recovery scheme ‘is based on maintaining a dual mapping between pages **and their locations on disk.**’” *Id.* at 20 (quoting Ex. 1008, 91 (Abstract)). Patent Owner further notes that Figure 4 of Lorie, which Petitioner cites for teaching details of the claimed “file management contexts,” shows vector V_{10} (corresponding to System R’s current page table) pointing to slots of a disk segment. *Id.* (citing Ex. 1008, 98). In addition, Lorie discloses storing a modified current page in slot j' on disk, and updating the current page table to map the page to slot j' , while maintaining the shadow page at slot j . Ex. 1008, 97; *see* Prelim. Resp. 20.

Patent Owner also identifies an apparent deficiency in Petitioner’s argument that a person of ordinary skill in the art would have understood that the current page table in System R tracks pages in non-persistent memory that do not survive restart. Prelim. Resp. 20 (citing Pet. 28–35). Patent Owner again points to Lorie, which, according to Patent Owner,

explains that the reason the current version of a page does not survive restart is not that it is stored in memory. *Id.* at 20–21 (citing Ex. 1008, 98). Rather, Lorie explains that in the event of system failure, a disk segment that is open (because some pages are being modified in the page buffer) lacks consistency and integrity, and therefore upon restart the system is restored to its last integrity state (i.e., the shadow version of pages in the disk segment rather than the current version). *See id.* at 21 (citing Ex. 1008, 98).

The Petition does not attempt to reconcile these various teachings in Lorie with Petitioner’s assertion that the current page table tracks changed pages *in non-persistent memory* rather than on disk (i.e., persistent storage). *See* Pet. 28–35. Therefore, on the present record, we are not persuaded that Petitioner has made a sufficient showing that System R’s current page table tracks changed pages *in memory*, as Petitioner contends, rather than changed pages *on disk*. Consequently, because Petitioner’s argument that the current page table resides in memory depends on its assertion that the current page table tracks pages in memory, Petitioner’s failure to make a sufficient showing regarding that assertion further undercuts its position.

For at least these reasons, based on the record at this stage of the proceeding, Petitioner has not shown sufficiently that the combination of System R and Lorie teaches or suggests a “file management context” stored in “non-persistent memory,” as required by independent claims 1, 2, and 4. Accordingly, we determine that the information presented does not show a reasonable likelihood that Petitioner would prevail in establishing that any of independent claims 1, 2, and 4, or dependent claims 5–8, 11, 14, 17–19, 21, 22, and 28, would have been obvious over the combined teachings of System R and Lorie.

E. Remaining Asserted Obviousness Grounds

Petitioner contends that dependent claims 9, 10, 12, 15, 20, and 24–27 are unpatentable under 35 U.S.C. § 103(a) as obvious over the combination of System R, Lorie, and Astrahan, and that dependent claim 23 is unpatentable under 35 U.S.C. § 103(a) as obvious over the combination of System R, Lorie, and R*. Pet. 70–86. Petitioner cites Astrahan and R* for teaching additional limitations recited in these dependent claims, not for teaching a “file management context” in “non-persistent memory.” *Id.* Therefore, Petitioner’s contentions do not address the deficiencies identified above with respect to independent claims 1, 2, and 4. Accordingly, we determine that the information presented does not show a reasonable likelihood that Petitioner would prevail in establishing that claims 9, 10, 12, 15, 20, and 23–27 would have been obvious over the asserted combinations of references.

IV. CONCLUSION

After considering the evidence and arguments presented in the Petition and Preliminary Response, we determine that the information presented does not show a reasonable likelihood that Petitioner would prevail in establishing that at least one of claims 1, 2, 4–12, 14, 15, and 17–28 of the ’419 patent is unpatentable on the grounds asserted in the Petition.

V. ORDER

Accordingly, it is

ORDERED that the Petition is *denied*, and no *inter partes* review is instituted.

IPR2021-01017
Patent 7,904,419 B1

PETITIONER:

Matthew W. Johnson
Joseph M. Beauchamp
JONES DAY
mwjohnson@jonesday.com
jbeauchamp@jonesday.com

Andrew M. Mason
KLARQUIST SPARKMAN, LLP
andrew.mason@klarquist.com

J. Steven Baughman
PAUL, WEISS, RIFKIND, WHARTON & GARRISON LLP
sbaughman@paulweiss.com

PATENT OWNER:

Mehran Arjomand
Akira Irie
Shouvik Biswas
Roman Swoopes
Fahd H. Patel
MORRISON & FOERSTER LLP
marjomand@mofo.com
airie@mofo.com
sbiswas@mofo.com
rswoopes@mofo.com
fpatel@mofo.com

David L. Fehrman
DSA LEGAL SOLUTIONS PC
dfehrman@dsa-legal.com