

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SAP SE,
Petitioner,

v.

TERADATA US, INC.,
Patent Owner.

IPR2021-00855
Patent 6,763,357 B1

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

PETTIGREW, *Administrative Patent Judge*.

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

I. INTRODUCTION

Petitioner, SAP SE, filed a Petition for *inter partes* review of claims 1–7, 26–33, and 52–57 of U.S. Patent No. 6,763,357 B1 (Ex. 1001, “the ’357 patent”). Paper 2 (“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 shows a reasonable likelihood that Petitioner would prevail in establishing the unpatentability of at least one of the challenged claims of the ’357 patent. Accordingly, we institute an *inter partes* review of claims 1–7, 26–33, and 52–57 of the ’357 patent on the grounds asserted in the Petition.

II. BACKGROUND

A. Real Parties-in-Interest

Petitioner identifies SAP SE, SAP America, Inc., and SAP Labs, LLC as real parties-in-interest. Pet. 92. 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 ’357 patent: *Teradata US, Inc. v. SAP SE*, Case No. 3:20-cv-06127

(N.D. Cal.), filed August 31, 2020 (the “parallel proceeding”). Pet. 92; Paper 4, 1.

C. Overview of the '357 Patent

The '357 patent describes an “active cache” for caching multi-dimensional data sets for an on-line analytical processing (OLAP) system that uses a relational database management system (RDBMS). Ex. 1001, 1:65–2:2. The active cache “can not only answer queries that match data stored in the cache, but can also answer queries that require aggregation or other computation of the data stored in the cache.” *Id.* at 3:2–5.

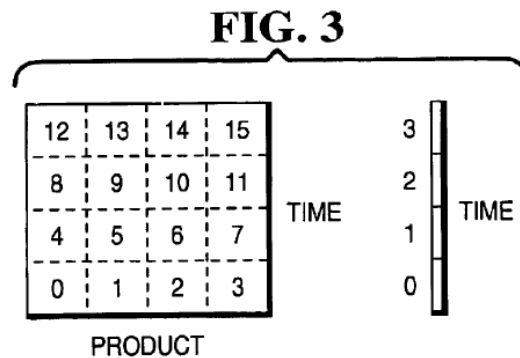
The background section of the '357 patent explains that OLAP systems provide tools for analysis of multi-dimensional data. *Id.* at 2:12–13. According to the '357 patent, most OLAP systems are built using a three-tier architecture: a first tier providing a graphical user interface, a second tier providing a multi-dimensional view of the data, and a third tier comprising an RDBMS that stores the data. *Id.* at 2:14–20. The '357 patent notes that “[m]ost queries in OLAP systems are complex and require the aggregation of large amounts of data” and that “decision support applications in OLAP systems need to be interactive and demand fast response times.” *Id.* at 2:21–24. The '357 patent identifies various prior art techniques for speeding up queries that have been studied and implemented, including caching in the second tier. *Id.* at 2:24–47.

The '357 patent describes in some detail a “chunk-based caching” method proposed in Deshpande '98,¹ a conference paper authored by the

¹ Prasad M. Deshpande et al., *Caching Multidimensional Queries Using Chunks*, SIGMOD '98: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, 259–70, June 1998 (Ex. 1002, “Deshpande '98”).

named inventors of the '357 patent and one of the prior art references asserted by Petitioner in this proceeding. *Id.* at 2:48–51, 5:60–6:45. As the '357 patent explains, “[t]he motivation for chunk-based caching is to allow later queries to take advantage of any data overlap in the cache . . . with stored results from previous queries, even if the answers for the later queries are not totally contained in the stored results from the previous queries.” *Id.* at 5:61–65.

Figure 3 of the '357 patent illustrates an example of a multidimensional space divided into chunks:



Id. at 3:18–20. Figure 3 above shows a multidimensional space formed by two dimensions, Product and Time, and divided into chunks at two different levels, (Product, Time) and (Time). *Id.* at 6:5–7.

The '357 patent explains that the chunk-based caching scheme described in Deshpande '98 uses chunks as a unit of caching. *Id.* at 6:7–8. Query results to be stored in the cache are broken up into chunks so that the chunks can be cached. *Id.* at 6:11–13. When a new query is issued, Deshpande '98's method analyzes the query to determine which chunks are needed to answer it and then probes the cache to determine whether those chunks are present in the cache. *Id.* at 6:13–15. The list of chunks needed to answer the query is divided into one partition comprising chunks that may

be answered from the cache and a second partition comprising missing chunks, which then may be computed from data obtained from the RDBMS. *Id.* at 6:16–28.

The '357 patent purports to improve upon Deshpande '98's method for computing missing chunks by providing a mechanism for aggregating data already contained in the cache rather than data in the RDBMS. *See id.* at 2:59–64, 4:8–9, 6:46–48. The “active cache” described in the '357 patent allows for such aggregation of data. *Id.* at 4:8–14. The '357 patent explains that its aggregation method addresses two related issues: (1) determining whether the data chunks in the cache provide a sufficient basis from which to compute the answer to the query, and (2) selecting the best path for aggregating data chunks from among multiple aggregation paths. *Id.* at 4:15–38.

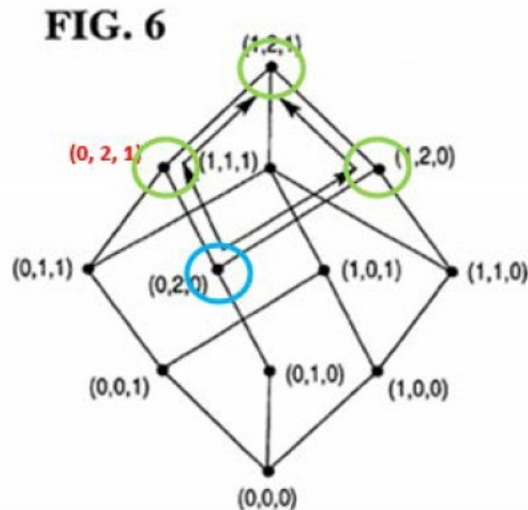
To illustrate the possibility of multiple aggregation paths, the '357 patent provides an example of a data schema having three dimensions—A, B, and C—as shown in Figure 5:

FIG. 5

DIMENSION	LEVELS
A	$A_1, A_0 = \text{ALL}$
B	$B_2, B_1, B_0 = \text{ALL}$
C	$C_1, C_0 = \text{ALL}$

Id. at 3:23–24. In the schema shown in Figure 5 above, dimension B has a two-level hierarchy, and dimensions A and C each have a single-level hierarchy. *Id.* at 6:56–59. Figure 6 (annotated by Petitioner) shows a lattice

formed by dimensions A, B, and C, where (x, y, z) denotes the level on each of the dimensions:



Pet. 11;² Ex. 1001, 3:25–26. Annotated Figure 6 above illustrates how data at one level in the cache (e.g., (1, 2, 1), or $A_1B_2C_1$, is the most detailed, granular level) can be aggregated to answer queries at a more aggregated, less granular level (e.g., (0, 0, 0), or $A_0B_0C_0$, is the most aggregated). See Ex. 1001, 6:59–62. For instance, data at level (0, 2, 0) can be computed from data at (0, 2, 1) or (1, 2, 0), if either is contained in the cache, using a “GROUP-BY” operation. *Id.* at 6:63–7:2. The ’357 patent explains that the problem becomes more complex when using Deshpande ’98’s chunk-based caching scheme. *Id.* at 7:3–18.

The ’357 patent describes two algorithms for determining whether a particular data chunk needed to answer a query can be computed from data found in the cache (i.e., to address the first issue described above): the Exhaustive Search Method (ESM), *id.* at 7:20–9:9, and the Virtual Count

² Petitioner’s annotated Figure 6 includes a correction to the original figure, shown in red. See Pet. 10 n.1 (citing Ex. 1001, 6:56–59; Ex. 1006 ¶ 26 (Lehner Decl.)).

Method (VCM), *id.* at 9:10–13:42. The '357 patent also describes two algorithms for selecting the best aggregation path (i.e., to address the second issue described above): cost-based ESM, *id.* at 13:64–15:29, and cost-based VCM, *id.* at 15:30–17:8.

D. Illustrative Claim

Challenged claims 1, 26, and 28 are independent; claims 2–7 depend directly or indirectly from claim 1; claims 27 and 52–57 depend directly or indirectly from claim 26; and claims 29–33 depend directly or indirectly from claim 28. Claim 1, reproduced below, is illustrative for purposes of the issues raised at this stage of the proceeding:

1. A method for caching multi-dimensional data sets for an on-line analytical processing (OLAP) system, comprising:
 - (a) accessing multi-dimensional data from a relational database management system (RDBMS) in order to answer at least a portion of a first query;
 - (b) storing the multi-dimensional data in an active cache;
 - (c) determining when it is possible to answer at least a portion of a second query by aggregating the multi-dimensional data stored in the cache; and
 - (d) selecting from among a plurality of aggregation paths and answering the portion of the second query by aggregating the multi-dimensional data stored in the cache using the selected aggregation path, after it has been determined that it is possible to answer the portion of the second query by aggregating the multi-dimensional data stored in the cache.

Ex. 1001, 28:37–53.

E. Asserted Grounds of Unpatentability

Petitioner asserts that claims 1–7, 26–33, and 52–57 are unpatentable based on the following grounds (Pet. 16):

Claims Challenged	35 U.S.C. §	Reference(s)
1–7, 26–33, 52–57	103(a) ³	Deshpande '98, Shim ⁴
1–5, 26–31, 52–55	103(a)	Albrecht 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 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

³ The Leahy-Smith America Invents Act, Pub. L. No. 112-29, 125 Stat. 284 (2011) (“AIA”), amended 35 U.S.C. § 103. Because the '357 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.

⁴ Junho Shim et al., *Dynamic Caching of Query Results for Decision Support Systems*, Eleventh International Conference on Scientific and Statistical Database Management, 254–63, July 1999 (Ex. 1003, “Shim”).

⁵ Jens Albrecht, *On-line Analytical Processing in Distributed Data Warehouses*, Proceedings IDEAS 1998, International Database Engineering and Applications Symposium, 78–85, July 1998 (Ex. 1004, “Albrecht”).

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 industry experience with database system development (or an equivalent degree or experience).” Pet. 14–15 (citing Ex. 1006 ¶ 22). Petitioner further contends that a person of ordinary skill in the art could have had “some combination of (1) more formal education (such as a master’s of science degree) and less technical experience, or (2) less formal education and more technical or professional experience.” *Id.* at 15 (citing Ex. 1006 ¶ 22).

Patent Owner does not dispute Petitioner’s proposed level of ordinary skill at this stage of the proceeding. Prelim. Resp. 9.

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 ’357 patent and the asserted prior art. *See Okajima v. Bourdeau*, 261 F.3d 1350, 1355 (Fed. Cir. 2001).

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).

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.

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 agrees that most terms do not require construction, but disagrees as to the term “active cache,” recited in independent claims 1, 26, and 28. Prelim. Resp. 9. Patent Owner proposes construing that term to mean a “cache that can not only answer queries that match data stored in the cache, but can also answer queries that require aggregation or other computation of the data stored in the cache,” in accordance with the ’357 patent’s description of an “active cache.” *Id.* at 10; *see* Ex. 1001, 3:1–5. As Patent Owner points out, the Petition acknowledges that description of an “active cache” in the ’357 patent and appears to apply such a construction in its unpatentability analysis. Prelim. Resp. 9–10 (citing Pet. 35, 72).

In view of the ’357 patent’s description of an “active cache,” at this stage of the proceeding we adopt the proposed construction of “active cache” as a “cache that can not only answer queries that match data stored in the cache, but can also answer queries that require aggregation or other computation of the data stored in the cache.” During trial, if appropriate, the parties may further address the construction of this term.

At this stage of the proceeding, no other terms require express construction. *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 Deshpande '98 and Shim

Petitioner contends that claims 1–7, 26–33, and 52–57 of the '357 patent are unpatentable under 35 U.S.C. § 103(a) as obvious over Deshpande '98 and Shim, relying on the Declaration of Dr. Lehner for support. Pet. 31–68 (citing Ex. 1006). Patent Owner disagrees, arguing that the combination of Deshpande '98 and Shim does not teach every limitation of independent claims 1, 26, and 28. Prelim. Resp. 11–22.

1. Overview of Deshpande '98

Deshpande '98 describes a method for caching multidimensional data in an OLAP system in units referred to as “chunks.” Ex. 1002, 259 (Abstract). “Chunk-based caching allows fine granularity caching, and allows queries to partially reuse the results of previous queries with which they overlap.” *Id.* The chunk-based scheme “divid[es] the multidimensional query space uniformly into chunks and cach[es] these chunks” so that “[t]he results of a query are contained in an integral number of chunks, which form a ‘bounding envelop’ around the query result.” *Id.* at 260. “Since chunks are at a lower granularity than query level caching,” which caches an entire query result, the chunks “can be reused to compute the partial result of incoming queries.” *Id.*

In Deshpande '98's chunk-based caching scheme, query results are broken up into chunks, which are stored in the cache. *Id.* at 261. When a new query is issued, the disclosed method determines which chunks are needed to answer that query. *Id.* at 261, 264–65. Then, depending on the cache contents, the list of required chunks is split into two partitions—one

comprising chunks present in the cache and the other comprising chunks to be computed from the backend database. *Id.* at 261–62, 265.

Deshpande '98 further explains that “[c]ached chunks can be used for a query when they are at the same or lower level of aggregation than the query.” *Id.* at 265. However, because the problem of figuring out which chunks can be aggregated to get a result chunk is “quite complex,” Deshpande '98's current implementation reuses cached chunks to answer a portion of a query only if they are at the same level of aggregation as the query. *Id.* If any aggregations are required to compute missing chunks, they are performed in the backend database. *Id.* at 265, 270. The authors describe their “future work” as “explor[ing] the possibility of aggregating chunks in the cache to get a missing chunk rather than going to the backend.” *Id.* at 270.

2. Overview of Shim

Shim describes a dynamic query cache manager for a decision support system (DSS) environment that aims to minimize query response time. Ex. 1003, 1. Shim's cache manager “can lookup query results from the cache . . . using a *query split* algorithm to efficiently find query results which subsume the submitted query.” *Id.* When the cache manager receives a query in “canonical form” (i.e., a query with a standard structure), it transforms the query according to the split algorithm and examines the cache content for “query results that may subsume the transformed query.” *Id.* at 2; *see id.* at 3–4 (describing canonical form). “If several candidates are found, a selection is made so that the average query response time is minimized.” *Id.* at 2.

Figure 2 of Shim provides an example of a canonical query, Q1, and data cubes DC1 and DC2 in the cache that could be used to respond to Q1:

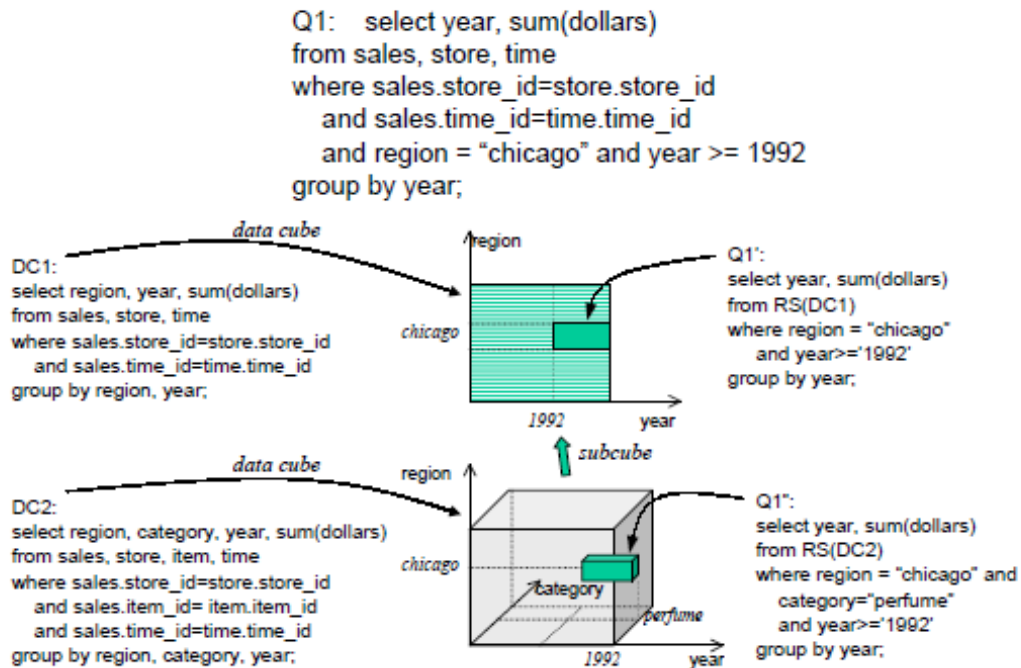


Figure 2: A canonical query and data cubes

Id. at 3. Figure 2 above shows that Shim’s cache manager may respond to query Q1 using data cubes, such as DC1 and DC2, containing data cached from previous queries. *Id.* Shim explains that “[d]ata cubes are related in a

derivability relation that forms a lattice structure.” *Id.* at 4. Figure 3 of Shim illustrates a lattice of data cubes that would be relevant to Q1:

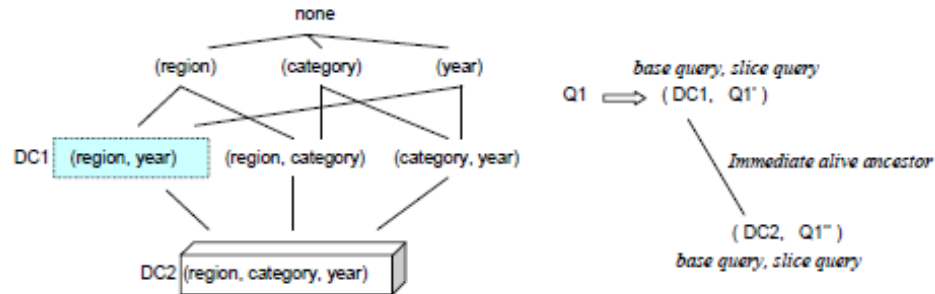


Figure 3: An example lattice and base, IA, and slice queries

Id. Figure 3 above shows a lattice structure including data cubes DC1 and DC2 from Figure 2. *Id.*

When Shim’s cache manager receives a query in canonical form, a “data cube whose cells contain the aggregations over all the attributes appearing in the *select-condition* of [the] query can be used for the evaluation of the query” if it is materialized in the cache. *Id.* But “if the corresponding data cube is not yet materialized in the cache, the lattice structure can be exploited to determine its immediate alive ancestor(IA) so that the given query can be evaluated.” *Id.* An immediate alive ancestor (IA) is a data cube “that was already materialized in the cache and there is no other materialized ancestor between the given data cube and *IA* along the lattice hierarchy.” *Id.* In Figure 3, for example, “if neither (region, year) nor (region, category) are materialized but (region, category, year) is found in the cache, (region, category, year) is the immediate alive ancestor of (region).” *Id.*

As detailed in Figure 2, query Q1 requests the sum of sales in the Chicago region for years 1992 and later. *Id.* at 3. Data cube DC1, which contains data from prior queries that requested the sum of sales for regions including Chicago for multiple years including 1992 and later, is a “data cube whose cells contain the aggregations over all the attributes appearing in the *select-condition* of” Q1. *Id.* at 4. If DC1 is materialized in the cache, Shim’s query split algorithm can transform Q1 into (DC1, Q1’) and return a result set from DC1 with the sum of sales in Chicago for 1992 and later, as requested by Q1. *Id.* But if DC1 is not materialized in the cache, “the split algorithm will select an immediate alive ancestor with the least estimated cost that can be used to evaluate” Q1. *Id.* For example, if DC2 is materialized in the cache and determined to be the IA of Q1 with the least estimated cost, a result set is obtained from DC2 by executing Q1”. *Id.* Shim explains that “[f]or simplicity, [the authors] assume that the execution cost of *Q* on a data cube (or raw data) is proportional to the number of rows scanned in the data cube (or raw data).” *Id.*

3. Independent Claim 1

a. Preamble and Limitation 1(a)

Claim 1 recites “[a] method for caching multi-dimensional data sets for an on-line analytical processing (OLAP) system.” Ex. 1001, 28:37–39. To the extent this preamble is limiting, Petitioner contends that Deshpande ’98 discloses a method for caching multi-dimensional data sets (in the form of chunks) for an OLAP system. Pet. 31 (citing Ex. 1002, 259 (Abstract)).

Limitation (a) of claim 1 recites “accessing multi-dimensional data from a relational database management system (RDBMS) in order to answer at least a portion of a first query.” Ex. 1001, 28:40–42. Petitioner contends

that Deshpande '98 describes a three-tier architecture for facilitating interactions with an RDBMS. Pet. 31–32 (citing Ex. 1002, 259 (“The middle tier provides a multidimensional view of the data stored in the final tier, which can be a RDBMS.”)). Petitioner also cites Deshpande '98's chunk-based caching methodology, which may answer at least a portion of a query by accessing multi-dimensional data from the RDBMS. *Id.* at 32–34 (citing Ex. 1002, 259–60, 262). Petitioner relies on Figure 2 of Deshpande '98, reproduced below:

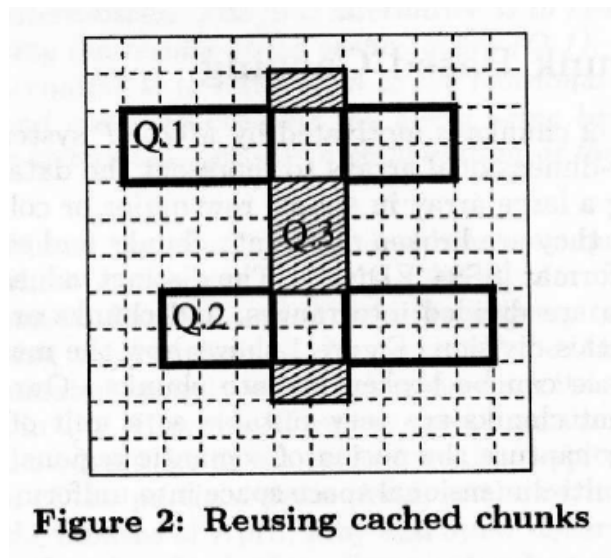


Figure 2: Reusing cached chunks

Ex. 1002, 262; *see* Pet. 34. Petitioner points to query Q.1 in Figure 2 above as an example of a query that is answered using multi-dimensional data from the RDBMS, which is stored as chunks in the cache. Pet. 32–33. At least a portion of a later query, Q.3, may be answered using the stored chunks, while at least a portion of Q.3 (i.e., any chunk missing from the cache) may be answered using data in the RDBMS. *Id.* at 33.

Patent Owner does not present arguments regarding the preamble or limitation 1(a). *See* Prelim. Resp. 11–22. At this stage of the proceeding

and on the present record, Petitioner's cited evidence sufficiently supports its contentions regarding the preamble and limitation 1(a).⁷

b. Limitations 1(b), 1(c), and 1(d)

Limitation (b) of claim 1 recites “storing the multi-dimensional data in an active cache.” Ex. 1001, 28:43. For this limitation, Petitioner first relies on Deshpande '98 for teaching storing the multi-dimensional data in a cache (e.g., the data chunks from the Q.1 query results in the cache shown in Figure 2). Pet. 34–35 (citing Ex. 1002, 262, Fig. 2).

Limitation (b) also requires an “active cache,” construed at this stage of the proceeding as a “cache that can not only answer queries that match data stored in the cache, *but can also answer queries that require aggregation or other computation of the data stored in the cache.*” *Supra* § III.C (emphasis added). Similarly, limitation (c) recites “determining when it is possible to answer at least a portion of a second query *by aggregating the multi-dimensional data stored in the cache,*” and limitation (d) recites in part “answering the portion of the second query *by aggregating the multi-dimensional data stored in the cache.*” Ex. 1001, 28:44–46, 28:48–50 (emphases added). Petitioner asserts that Deshpande '98 “previews” an “active cache” that allows for in-cache aggregation in its “future work” section: “In our current implementation, all aggregations are restricted to the backend. For future work, we are planning to explore the possibility of aggregating chunks in the cache to get a missing chunk rather than going to the backend.” Ex. 1002, 270, *quoted in* Pet. 36; *see also* Pet. 43 (in limitation (c) analysis, asserting that “Deshpande '98

⁷ Because we determine Petitioner sufficiently shows that Deshpande '98 teaches the subject matter in the preamble, we need not decide whether the preamble is limiting at this stage of the proceeding.

previews in-cache aggregation”). Petitioner acknowledges, however, that Deshpande ’98 does not include details of how to implement in-cache aggregation when answering queries. Pet. 36.

To fill in those details, Petitioner turns to Shim. *Id.* at 36–38 (limitation (b)), 43–50 (limitation (c)), 51–54 (limitation (d)). In particular, Petitioner contends that Shim teaches “a cache configuration where data cached at a finer granularity can be aggregated to answer a query at a higher aggregation level.” *Id.* at 36. According to Petitioner, “Shim takes advantage of the same lattice structure that the ’357 patent exploits to answer queries requesting data at a higher level of aggregation than currently cached data.” *Id.* (comparing Shim’s Figure 3 with annotated Figure 6 of the ’357 patent). Petitioner’s comparison of Shim’s Figure 3 and annotated Figure 6 of the ’357 patent is reproduced below:

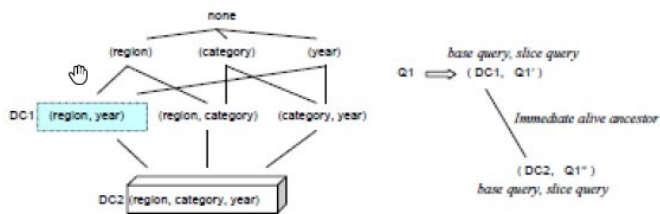
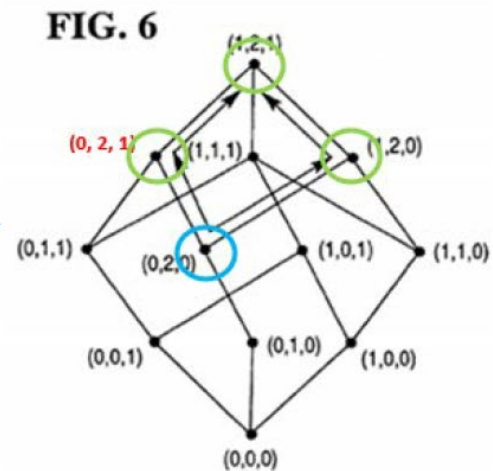


Figure 3: An example lattice and base, IA, and slice queries



As shown above, Petitioner asserts that Shim’s Figure 3 and Figure 6 of the ’357 patent (annotated by Petitioner) show the same lattice structure.

Pet. 37. More specifically, Petitioner contends that

when Shim receives a query at the (region) level of its lattice hierarchy and sufficient data is not cached at that level, Shim searches lower levels of the lattice hierarchy for an immediately

alive ancestor (IA) containing data that can be aggregated to achieve an answer for the (region) level query. . . . Shim discloses that when a “data cube whose cells contain the aggregations over all the attributes appearing in the select-condition of a query . . . is not yet materialized in the cache, the lattice structure can be exploited to determine its immediate alive ancestor(IA) so that the given query can be evaluated on the found IA.”

Id. (quoting Ex. 1003, 4; citing Ex. 1001, 13:61–63 (’357 patent aggregating from “immediate ancestor”)).

Limitation (c) also requires “*determining when it is possible to answer at least a portion of a second query* by aggregating the multi-dimensional data stored in the cache.” Ex. 1001, 28:44–46 (emphasis added). Petitioner contends that in Shim, when the data needed to answer a query “is not present at the query-requested aggregation level,” Shim’s split algorithm “traverses the lattice, level by level, until at least one data cube of cached data is found.” Pet. 45; *see* Ex. 1003, 4, Fig. 4 (providing details of the split algorithm). “If at least one cached ancestor data cube has been found, then the algorithm will progress . . . to select the most efficient ancestor from which to aggregate data to answer the query.” Pet. 48. But if no ancestor data cubes are found in the cache, Petitioner contends, “the algorithm would thus direct the system to access the needed data from the underlying database because that needed data is not currently stored in the cache.” *Id.* at 47; *see* Ex. 1006 ¶ 114. In this way, Petitioner continues, “Shim’s algorithm searches the lattice at the query . . . level of aggregation granularity and finer to *determine whether there are any data cubes materialized in the cache from which the query . . . can be answered.*” Pet. 49 (emphasis added). Thus, Petitioner concludes, Deshpande ’98 in view of Shim renders obvious limitation (c) of claim 1. *Id.* at 50.

Limitation (d) recites

selecting from among a plurality of aggregation paths and answering the portion of the second query by aggregating the multi-dimensional data stored in the cache using the selected aggregation path after it has been determined that it is possible to answer the portion of the second query by aggregating the multi-dimensional data stored in the cache.

Ex. 1001, 28:47–53 (emphasis added). Petitioner contends that when Shim’s split algorithm finds multiple ancestor data cubes in the cache, it “selects ‘the data cube with the least estimated cost to evaluate [the query] Q using it.’” Pet. 53 (quoting Ex. 1003, 4). Shim provides that “the execution cost of Q on a data cube . . . is proportional to the number of rows scanned in the data cube.” Ex. 1003, 4, *quoted in* Pet. 53. Petitioner asserts that the ’357 patent similarly tracks aggregation costs. Pet. 53–54 (citing Ex. 1001, 13:46–48 (“A linear cost of aggregation is assumed, i.e., the cost is proportional to the number of tuples aggregated.”)). Thus, Petitioner concludes, Deshpande ’98 in view of Shim renders obvious limitation (d) of claim 1. *Id.* at 54.

Petitioner also provides a rationale for the combination, arguing that a person of ordinary skill in the art

would have been motivated to combine the teachings of Deshpande ’98 and Shim for a variety of reasons, most notably because Deshpande ’98 explicitly states that in-cache aggregation is the next logical step in multi-dimensional cache development, and Shim advantageously provides an implementation of that aggregation suggested by Deshpande ’98.

Pet. 38. Thus, Petitioner contends, a person of ordinary skill in the art “would have been explicitly directed to look for opportunities to improve on Deshpande ’98’s system via in-cache aggregation,” and that direction would

have naturally led to Shim. *Id.* at 39 (citing Ex. 1006 ¶ 98). Relying on Dr. Lehner’s testimony for support, Petitioner contends that “[a]ccessing data from cache memory is well known to be significantly faster than accessing data from a database,” “[i]n-cache aggregation [as taught in Shim] increases the amount of cached data that is eligible for answering a current query,” “[m]ore eligible cache data . . . results in more cache hits,” and “a greater number of cache hits results in faster average performance.” *Id.* at 39–40 (citing Ex. 1006 ¶¶ 99–100). Petitioner also notes that Shim cites to Deshpande ’98. *Id.* at 41 (citing Ex. 1003, 10). Finally, Petitioner contends that a person of ordinary skill in the art would have had a reasonable expectation of success in implementing the combination of Shim and Deshpande ’98 in view of Shim’s example descriptions and pseudocode algorithms. *Id.* (citing Ex. 1006 ¶ 102).

Patent Owner argues that “Petitioner fails to present any evidence that Shim *or* Deshpande ’98 disclose[s] ‘determining when it is possible to answer at least a portion of a second query by aggregating the multi-dimensional data stored in the cache,’” as recited in limitation 1(c), and similarly “fails to explain how Deshpande ’98 *or* Shim discloses or renders obvious ‘answering the portion of the second query by aggregating the multi-dimensional data stored in the cache,’” as recited in limitation 1(d). Prelim. Resp. 11, 22 (emphases added); *see also id.* at 14–18 (arguing that Shim does not teach answering a portion of a query), 18–19 (arguing that Deshpande ’98 does not teach answering a portion of a query by aggregating cached data). As an initial matter, Petitioner relies on the combination of Deshpande ’98 and Shim for teaching these limitations. Pet. 50, 54 (asserting that “Deshpande ’98 in view of Shim renders obvious” limitations 1(c) and 1(d)). Therefore, to the extent Patent Owner’s argument

attacks the references individually, it is unpersuasive. *See In re Keller*, 642 F.2d 413, 426 (CCPA 1981).

Patent Owner takes issue with Petitioner’s contention that “Shim teaches the entirety of [limitation 1(c)]: determining when it is possible to answer at least a portion (*e.g.*, all) of a second query by aggregating the multi-dimensional data stored in the cache.” Prelim. Resp. 11 (quoting Pet. 43). Patent Owner contends that “Petitioner replaces the phrase ‘at least a portion’ with ‘all,’” and that the correct reading of limitation 1(c) “is that the database system in its analysis needs to be *capable of determining when it is possible to answer a portion of a second query* by aggregating multi-dimensional data stored in cache.” *Id.* Similarly, Patent Owner argues that Petitioner “re-writes ‘answering the portion of the second query’ in [limitation 1(d)] with ‘answering the portion (all) of the second query.’” *Id.* at 22 (quoting Pet. 54).

At this stage of the proceeding, and based on the present record, Patent Owner’s arguments do not identify a deficiency in Petitioner’s challenge. First, neither party proposes an express construction of “at least a portion of a second query” in limitation 1(c) or “the portion of the second query” in limitation 1(d). *See* Pet. 16–17; Prelim. Resp. 9–10. The parties may further address the construction of these claim phrases during trial.

Furthermore, even if the claim language requires answering only a portion (*i.e.*, less than all) of a second query, as Patent Owner seems to contend, on this preliminary record we find that Petitioner has made a sufficient showing supported by evidence. Importantly, Petitioner relies on the combined teachings of Deshpande ’98 and Shim for teaching limitations 1(c) and 1(d). Pet. 42–55. Petitioner begins its analysis of limitation 1(c) by citing Deshpande ’98’s teaching that the list of chunks

needed to answer a new query is partitioned into one portion that is answered using chunks in the cache and a second portion, consisting of “missing chunks,” that has to be computed from the backend database. *Id.* at 42–43 (citing Ex. 1002, 261–62). Petitioner notes that Deshpande ’98 “previews” the possibility of aggregating chunks in the cache to obtain the missing chunks needed to answer the second portion of the query, and Petitioner then turns to Shim for teaching in-cache aggregation. *Id.* at 36, 43 (citing Ex. 1002, 270; Ex. 1003, 4). Thus, in its proposed combination of references, Petitioner appears to rely on Shim’s in-cache aggregation to compute Deshpande ’98’s missing chunks. *See id.* at 36, 43. In other words, at least based on the present record, the cited evidence supports a showing that the proposed combination of Deshpande ’98 and Shim teaches or suggests answering a portion of a query (i.e., Deshpande ’98’s missing chunks) by aggregating multi-dimensional data stored in the cache, as taught by Shim.

Patent Owner also argues that no motivation exists for combining Shim with Deshpande ’98 because “Shim completely ignores the chunk-based approach that [Deshpande ’98] describes” and chose a completely different approach even though “Shim was fully aware of Deshpande ’98.” Prelim. Resp. 19. According to Patent Owner, Petitioner paints a “broad brush . . . over the significant differences that exist between [Shim’s] data cubes and [Deshpande ’98’s] chunks.” *Id.* at 20. Patent Owner further argues that the ’357 patent explains that “the cache aggregation ‘problem becomes more complex when using the chunk-based caching scheme,’ [and] [t]he Petition offers no explanation of how one skilled in the art would overcome these difficulties when trying to apply teachings from Shim’s data cube-based system to Deshpande ’98’s chunk-based system.” *Id.* (quoting

Ex. 1001, 7:3–4). Notwithstanding Patent Owner’s arguments, we determine that the Petition, supported by the testimony of Petitioner’s declarant, Dr. Lehner, provides sufficient reasoning for combining the references at this stage of the proceeding. Patent Owner will have the opportunity to develop any contrary arguments and evidence further at trial.

For these reasons, at this stage of the proceeding, Petitioner provides sufficient evidence to support its contentions regarding limitations 1(b), 1(c), and 1(d).

c. Conclusion as to Claim 1

We have considered the arguments and supporting evidence presented by the parties. For the reasons discussed above, based on the totality of the evidence currently in the record, we are persuaded that Petitioner has shown sufficiently at this stage of the proceeding that the combined teachings of Deshpande ’98 and Shim teach or suggest each limitation of claim 1 and that a person of ordinary skill in the art would have had reason to combine the references in the manner Petitioner proposes with a reasonable expectation of success. Therefore, we determine that the information presented shows a reasonable likelihood that Petitioner would prevail in establishing that claim 1 would have been obvious over Deshpande ’98 and Shim.

4. Independent Claims 26 and 28

Independent claim 26 is a system claim that recites a computer system with logic for performing the same steps recited in claim 1, and independent claim 28 is an article of manufacture claim that recites logic for performing the steps of claim 1. Ex. 1001, 30:7–25, 30:30–46. Petitioner argues these claims would have been obvious over Deshpande ’98 and Shim for the same reasons provided for claim 1. Pet. 67–68. Patent Owner presents the same arguments in response as those for claim 1. Prelim. Resp. 11–22.

For the same reasons discussed above with respect to claim 1, we determine that the information presented shows a reasonable likelihood that Petitioner would prevail in establishing that claims 26 and 28 are unpatentable over Deshpande '98 and Shim.

5. Dependent Claims

Petitioner contends that dependent claims 2–7, 27, 29–33, and 52–57, each of which depends from a challenged independent claim, would have been obvious over the combined teachings of Deshpande '98 and Shim. Pet. 55–68. Patent Owner does not provide any separate argument for these claims. *See* Prelim. Resp. 11–22. Based on the record at this stage of the proceeding, we determine that the information presented shows a reasonable likelihood that Petitioner would prevail in establishing that claims 2–7, 27, 29–33, and 52–57 are unpatentable over Deshpande '98 and Shim.

E. Asserted Obviousness over Albrecht

Petitioner contends that claims 1–5, 26–31, and 52–55 of the '357 patent are unpatentable under 35 U.S.C. § 103(a) as obvious over Albrecht. Pet. 69–89. Patent Owner disagrees, arguing that Albrecht does not teach the “cache” limitations of independent claims 1, 26, and 28. Prelim. Resp. 23–27. Because we determine that the information presented in the Petition and Preliminary Response demonstrates a reasonable likelihood that Petitioner would prevail in showing that these claims are unpatentable over the combination of Deshpande '98 and Shim, we need not address Petitioner's challenge based on Albrecht at this stage of the proceeding. *See* 35 U.S.C. § 314(a). Nonetheless, we offer the following observations regarding this challenge.

Albrecht describes an OLAP system having a three-tier architecture. Ex. 1004, 5, Fig. 6. End user tools with reporting facilities reside in the first

(client) layer. *Id.* “The heart of the architecture is the *middleware layer* hiding the details of data distribution. Its main task is to translate a user query and generate an optimized distributed query execution plan.” *Id.* The third (server) layer comprises a distributed data warehouse made up of a federation of distributed “data marts.” *Id.* at 1, 5, Fig. 6. The data marts contain raw data in the form of Base Multidimensional Objects (BMOs), which represent data at the finest granularity level. *Id.* at 6. The data marts also store Multidimensional Objects (MOs), which are partial aggregates containing data aggregated a higher level than BMOs that may be used to answer queries. *Id.* at 5–6. Albrecht discloses a “patchworking algorithm” for “finding the least expensive aggregate partitions.” *Id.* at 4.

In its asserted obviousness ground based on Albrecht, Petitioner contends that Albrecht discloses storing MOs (i.e., multidimensional data) in an “active cache,” as recited in independent claims 1, 26, and 28. Pet. 73. In response, Patent Owner argues that Albrecht’s MOs reside at the data marts, which Patent Owner contends “are the same as a backend database in a three-tiered database system—not the cache found in the middle layer.” Prelim. Resp. 23 (comparing Ex. 1004, 5, Fig. 6 (Albrecht’s third layer comprising a federation of data mart servers), with Ex. 1001, 4:49–52 (the ’357 patent’s third tier comprising a RDBMS)). Although “the word ‘cache’ appears in Albrecht 5 times,” Patent Owner argues, “the Petition cites none of these excerpts.” *Id.* (citing Pet. 73–74).

Contrary to Patent Owner’s argument, the Petition does cite certain references to a “cache” in Albrecht. For instance, the Petition quotes the following statement from Albrecht: “[A] self-maintaining, dynamic aggregate management working like a (distributed) system buffer or cache for aggregates offers the needed flexibility and the potential for high query

response times and low human maintenance efforts.” Ex. 1004, 3, *quoted in* Pet. 70. The Petition also provides the following quote from Albrecht: “[S]toring MOs that were already needed is working like a cache, with a microeconomic heuristics for replacement.” Ex. 1004, 7, *quoted in* Pet. 70. Petitioner apparently contends that, in view of these passages, Albrecht’s disclosure of storing MOs in data marts teaches or suggests storing multidimensional data in an active cache. *See* Pet. 70–73; Ex. 1006 ¶¶ 161–172. During trial, Patent Owner will have the opportunity to address whether Petitioner’s arguments and cited evidence sufficiently show that Albrecht teaches or suggests the limitations of the challenged claims, including those relating to storing multi-dimensional data in an active cache and aggregating the data stored in the cache.

*F. Discretion Under 35 U.S.C. § 314(a) Based on
Related District Court Proceeding*

In the Preliminary Response, Patent Owner contends that we should exercise our discretion to deny the Petition in favor of the parallel proceeding, which is scheduled for trial in January 2023. Prelim. Resp. 27–30. The Board has held that the advanced state of a parallel district court action is a factor that may weigh in favor of denying a petition under § 314(a). *See NHK Spring Co. v. Intri-Plex Techs., Inc.*, IPR2018-00752, Paper 8 at 20 (PTAB Sept. 12, 2018) (precedential) (“*NHK*”); Patent Trial and Appeal Board Consolidated Trial Practice Guide (Nov. 2019), 58 & n.2, *available at* <https://www.uspto.gov/TrialPracticeGuideConsolidated> (“Trial Practice Guide”). We consider the following factors to assess “whether efficiency, fairness, and the merits support the exercise of authority to deny institution in view of an earlier trial date in the parallel proceeding”:

1. whether the court granted a stay or evidence exists that one may be granted if a proceeding is instituted;
2. proximity of the court's trial date to the Board's projected statutory deadline for a final written decision;
3. investment in the parallel proceeding by the court and the parties;
4. overlap between issues raised in the petition and in the parallel proceeding;
5. whether the petitioner and the defendant in the parallel proceeding are the same party; and
6. other circumstances that impact the Board's exercise of discretion, including the merits.

Apple Inc. v. Fintiv Inc., IPR2020-00019, Paper 11 at 5–6 (PTAB Mar. 20, 2020) (precedential) (“*Fintiv*”). In evaluating these factors, we “take[] a holistic view of whether efficiency and integrity of the system are best served by denying or instituting review.” *Id.* at 6.

We consider each of these factors below.

1. Factor 1 — Stay of Related Litigation Proceeding

It appears that neither party has requested a stay of the related district court proceeding. Prelim. Resp. 27. Given the particular circumstances of this case, we determine that it would be improper to speculate, at this stage, what the district court might do regarding a motion to stay, should one be requested. *See Sand Revolution II, LLC v. Cont'l Intermodal Grp. – Trucking LLC*, IPR2019-01393, Paper 24 at 7 (PTAB June 16, 2020) (informative) (“*Sand Revolution*”) (“In the absence of specific evidence, we will not attempt to predict how the district court in the related district court litigation will proceed because the court may determine whether or not to stay any individual case, including the related one, based on a variety of circumstances and facts beyond our control and to which the Board is not

privy.”). Accordingly, we determine that this factor does not weigh either for or against exercising our discretion to deny institution in this case.

2. Factor 2 — Proximity of the Court’s Trial Date

Patent Owner asserts that the trial date in the parallel proceeding is scheduled for January 2023, but acknowledges that “the Board will have until November 20, 2021 to issue its decision on whether to institute the IPR.” Prelim. Resp. 27–28. “Because this is more than one year before the trial date,” Patent Owner states, “this factor weighs in favor of Petitioner.”

Any final written decision in this case will be issued no later than November 2022, which is approximately two months before the projected trial date in the parallel proceeding. Because the projected trial date is after the date for a final written decision, we agree with Patent Owner that this factor weighs against exercising our discretion to deny institution.

3. Factor 3 — Investment in the Parallel Proceeding

Patent Owner argues that the parties have invested significant resources into the parallel proceeding, including infringement and invalidity contentions, the exchange of potential claim terms for construction, and the exchange of significant volumes of documents and source code. Prelim. Resp. 28. “On the other hand,” Patent Owner acknowledges, “the Court is not expected to issue any orders relating to the ’357 Patent until mid or late 2022, as the Markman hearing is set for January 14, 2022 and summary judgment motions are not due until August 25, 2022.” *Id.* at 28–29 (citing Ex. 1010, 3).

Petitioner asserts that the parallel proceeding “is still in its early stages.” Pet. 90. Additionally, Petitioner argues that the Petition was filed just over three months after infringement contentions in the parallel proceeding, and roughly a month after service of invalidity contentions. *Id.*

The parallel proceeding is still at a relatively early stage, and we are not persuaded that the actions that have occurred so far indicate the type of significant investment that would support exercising our discretion to deny institution. *See Sand Revolution*, Paper 24 at 11 (observing that “much work remains in the district court case as it relates to invalidity: fact discovery is still ongoing, expert reports are not yet due, and substantive motion practice is yet to come”). We also credit Petitioner’s diligence in filing its Petition approximately three months after receiving infringement contentions in the parallel proceeding. In light of these facts, we determine that this factor weighs against exercising our discretion to deny institution.

4. Factor 4 — Overlap With Issues Raised in the Parallel Proceeding

Patent Owner argues that “[s]ignificant overlap exists between the invalidity challenges raised in this petition” and the parallel proceeding. Prelim. Resp. 29. According to Patent Owner, “Petitioner has identified the same set of references as invalidating prior art references and set forth substantially the same invalidity grounds as those raised in this Petition.” *Id.* (citing Ex. 2003; Ex. 2004). Petitioner does not specifically address this factor. *See* Pet. 89–90.

Based on the present record, there appears to be significant overlap between this proceeding and the parallel proceeding. Therefore, we determine that this factor weighs in favor of exercising our discretion to deny institution.

5. Factor 5 — Commonality of Parties in the Parallel Proceedings

Patent Owner asserts that Petitioner and Patent Owner are both parties to the parallel proceeding. Prelim. Resp. 29. Petitioner does not address this factor. *See* Pet. 89–90. Based on the present record, we find that this factor weighs against exercising our discretion to deny institution.

6. Factor 6 — Other Circumstances

Patent Owner does not identify any other circumstances that impact the Board’s exercise of discretion. Prelim. Resp. 30.

7. Conclusion

As noted in *Fintiv*, we consider the above factors as part of “a holistic view of whether efficiency and integrity of the system are best served by denying or instituting review.” *Fintiv*, Paper 11 at 6. Taken together, the scheduling of the trial date after the due date for a final written decision, the relatively early stage of the parallel proceeding, and Petitioner’s diligence in filing the Petition indicate that this is not a case where exercising our discretion to deny institution would be appropriate. Thus, considering the *Fintiv* factors as part of a holistic analysis, we are not persuaded that the interests of the efficiency and integrity of the system would be best served by invoking our discretion under 35 U.S.C. § 314(a) to deny institution of a potentially meritorious petition.

IV. CONCLUSION

After considering the evidence and arguments presented in the Petition and Preliminary Response, we determine that the information presented shows a reasonable likelihood that Petitioner would prevail in establishing that at least one of the challenged claims of the ’357 patent is unpatentable. We therefore institute trial on all challenged claims and grounds raised in the Petition. *See PGS Geophysical AS v. Iancu*, 891 F.3d 1354, 1360 (Fed. Cir. 2018) (indicating that a decision whether to institute an *inter partes* review “require[s] a simple yes-or-no institution choice respecting a petition, embracing all challenges included in the petition”).

V. ORDER

Accordingly, it is

ORDERED that, pursuant to 35 U.S.C. § 314(a), an *inter partes* review is instituted for claims 1–7, 26–33, and 52–57 of the '357 patent on the unpatentability grounds asserted in the Petition; and

FURTHER ORDERED that pursuant to 35 U.S.C. § 314(c) and 37 C.F.R. § 42.4, notice is hereby given of the institution of a trial, which commences on the entry date of this decision.

IPR2021-00855
Patent 6,763,357 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
David L. Fehrman
MORRISON & FOERSTER LLP
marjomand@mofo.com
airie@mofo.com
sbiswas@mofo.com
rswoopes@mofo.com

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