

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AMERICAN AIRLINES, INC.,
AND
HILTON DOMESTIC OPERATING COMPANY, INC.,
Petitioners

v.

R2 SOLUTIONS LLC,
Patent Owner

Case IPR2023-00887
U.S. Patent No. 8,209,317

**PETITION FOR *INTER PARTES* REVIEW OF
U.S. PATENT NO. 8,209,317**

TABLE OF CONTENTS

I.	Introduction.....	1
II.	Precise Relief Requested	2
III.	Overview of the Subject Matter.....	2
	A. Overview of the '317 Patent.....	2
	B. Overview of the Prosecution History	5
	C. The Level of Ordinary Skill in the Art	6
	D. Claim Construction.....	7
IV.	Overview of the Prior Art	8
	A. Background and State of the Art	8
	B. Overview of Sareen	10
	C. Overview of Anick	12
	D. Overview of Aravamudan	15
	E. Overview of Kamvar	16
V.	Ground 1: Claims 1-11 are Obvious in view of Sareen and Anick.....	18
	A. Claim 1	18
	1. Preamble	18
	2. Database.....	19
	3. Query Reconstruction Server	21
	4. Full Query Better Representative of Entire Query	29
	B. Claim 2	30
	C. Claim 3	31

D.	Claim 4	33
E.	Claim 5	36
F.	Claim 6	37
G.	Claim 7	39
H.	Claim 8	40
	1. Preamble	40
	2. Storing in Memory.....	41
	3. Retrieving Data from the Memory	41
	4. Constructing Indexes	42
	5. Receiving a Partial Query.....	43
	6. Determining a Full Query.....	44
I.	Claim 9	44
J.	Claim 10	45
K.	Claim 11	46
VI.	GROUND 2: CLAIMS 12-14 ARE OBVIOUS IN VIEW OF SAREEN IN COMBINATION WITH ANICK AND ARAVAMUDAN.....	49
A.	Claim 12	49
B.	Claim 13	51
C.	Claim 14	52
VII.	GROUND 3: CLAIMS 15-18 ARE OBVIOUS IN VIEW OF SAREEN IN COMBINATION WITH KAMVAR, ARAVAMUDAN AND ANICK.	53
A.	Claim 15	53
	1. Preamble	53

2.	Storing in Memory.....	54
3.	Receiving a Partial Query.....	56
4.	Interpreting Each Word of the Partial Query as a Partial Word.....	56
5.	Identifying Matching Queries.....	57
6.	Selecting a Full Query	57
B.	Claim 16	58
C.	Claim 17	59
D.	Claim 18	59
VIII.	The Board Should Institute <i>Inter Partes</i> Review	60
IX.	Mandatory Notices Under 37 C.F.R. § 42.8	63
A.	Real Party-in-Interest.....	63
B.	Related Matters.....	63
C.	Lead and Backup Counsel	65
D.	Service Information.....	66
X.	Grounds for Standing.....	66
XI.	Conclusion	67

TABLE OF EXHIBITS

Exhibit	Description
Ex-1001	U.S. Patent No. 8,209,317 (“317 patent”)
Ex-1002	File History of U.S. Patent No. 8,209,317
Ex-1003	Declaration of Benjamin B. Bederson, Ph.D.
Ex-1004	Curriculum Vitae of Benjamin B. Bederson, Ph.D.
Ex-1005	File History of U.S. Patent No. 7,716,201
Ex-1006	File History of U.S. Patent No. 8,046,347
Ex-1007	U.S. Patent Application Publication 2007/0203890 to Sareen et al. (“Sareen”)
Ex-1008	U.S. Patent 6,947,930 to Anick et al. (“Anick”)
Ex-1009	U.S. Patent Application Publication 2007/0005563 to Aravamudan et al. (“Aravamudan”)
Ex-1010	U.S. Patent Application Publication No. 2005/0283468 to Kamvar et al. (“Kamvar”)
Ex-1011	Claim Construction Memorandum Opinion and Order, <i>R2 Solutions LLC v. American Airlines Inc.</i> , No. 4:22-cv-00353, Dkt. 58 (E.D. Tex. April 3, 2023)
Ex-1012	Anick, Peter, <i>Using Terminological Feedback for Web Search Refinement – a Log-Based Study</i> , Conference: SIGIR 2003: Proceedings of the 26 th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (January 2003)

Exhibit	Description
Ex-1013	Plaintiff's Opening Claim Construction Brief in <i>R2 Solutions LLC v. American Airlines Inc.</i> , No. 4:22-cv-00353 (E.D. Tex. Feb. 21, 2023)
Ex-1014	U.S. Patent 7,461,059 to Richardson <i>et al.</i>
Ex-1015	Peri Tarr et al., <i>Workshop on Multi-Dimensional Separation of Concerns in Software Engineering</i> , ICSE '00 Proceedings of the 22nd International Conference on Software Engineering 809 (2000), DOI: https://doi.org/10.1145/337180.337827
Ex-1016	Scheduling Order, <i>R2 Solutions LLC v. American Airlines Inc.</i> , No. 4:22-cv-00353, Dkt. 20 (E.D. Tex. Oct. 7, 2022)
Ex-1017	Plaintiff's Complaint for Patent Infringement, <i>R2 Sols. LLC v. Hilton Domestic Operating Company, Inc.</i> , No. 3:22-cv-02761, Dkt. 1 (N.D. Tex. Dec. 12, 2022)
Ex-1018	Plaintiff's Disclosure of Asserted Claims and Infringement Contentions, 4:22-cv-00353 (E.D. Tex. Oct. 13, 2022)
Ex-1019	Docket Navigator Report for U.S. Patent No. 8,209,317 (last updated April 27, 2023)
Ex-1020	<i>Yahoo to buy Overture for \$1.63 billion</i> , CNET, https://www.cnet.com/tech/tech-industry/yahoo-to-buy-overture-for-1-63-billion/ (July 2003)

TABLE OF AUTHORITIES

	Page(s)
CASES	
<i>Becton, Dickinson & Co. v. B. Braun Melsungen AG</i> , IPR2017-01586, Paper 8 (Dec. 15, 2017).....	63
<i>In re GPAC Inc.</i> , 57 F.3d 1573 (Fed. Cir. 1995).....	6
<i>Netnut Ltd., v. Bright Data Ltd.</i> , IPR2021-01492, Paper 12 (Mar. 21, 2022)	61
<i>R2 Sols. LLC v. Hilton Domestic Operating Company, Inc.</i> , No. 3:22-cv-02761 (N.D. Tex. Dec. 12, 2022).....	6
<i>R2 Solutions LLC v. American Airlines Inc.</i> , No. 4:22-cv-00353 (E.D. Tex. Oct. 7, 2022)	6
<i>Sand Revolution II, LLC v. Cont'l Intermodal Grp.-Trucking LLC</i> , IPR2019-01393, Paper 24 (June 16, 2020).....	61, 62
<i>Sotera Wireless, Inc. v. Masimo Corp.</i> , IPR2020-01019, Paper 12 (Dec. 1, 2020).....	62
<i>Volkswagen Group of Am., Inc., v. Arigna Tech. Ltd.</i> , IPR2021-01321, Paper 10 (Feb. 15, 2022)	61
STATUTES	
35 U.S.C. §§ 102(a), (b) and (e).....	2
35 U.S.C. § 103	2
35 U.S.C. § 325(d)	62

I. Introduction

The '317 patent purports to improve on prior art search engines by returning search results for partial queries submitted by a user, rather than requiring the user to enter a full search query. Contrary to the '317 patent's assertions, however, this purported improvement was known at the time '317 patent was filed. Sareen, for example, teaches a system and method for generating search results for a partial query, such as the first few characters entered by a user, rather than requiring the user to enter a full search term or query.

During prosecution of the parent application to the '317 patent, the applicant narrowed the purported improvement to merely processing partial queries that a user has submitted through a deliberate and conscious act rather than processing partial queries dynamically as the user enters the text. But this concept is insufficient to render the claims patentable, as it too was well known in the art. Indeed, Anick discloses a method for doing precisely this and was readily combinable with Sareen's system and methods to render obvious most of the claim limitations – and many of the claims in their entirety – of the '317 patent.

The other prior art references in this petition—Kamvar and Aravamudan—teach the remaining elements recited in the '317 patent claims that are not rendered obvious by Sareen and Anick alone. The Board should institute *inter partes* review and cancel claims 1-18 of the '317 patent.

II. Precise Relief Requested

Petitioners request review and cancellation of claims 1-18 of the '317 patent as unpatentable in view of the following prior art:

Prior Art
U.S. Patent Publication No. 2007/0203890 to Sareen et al. ("Sareen"); filed February 28, 2006; published August 30, 2007; prior art under pre-AIA 35 U.S.C. § 102(e)
U.S. Patent No. 6,947,930 to Anick et al. ("Anick"); filed April 25, 2003 and issued September 20, 2005; prior art under pre-AIA 35 U.S.C. §§ 102(a) and (e)
U.S. Patent Publication No. 2007/0005563 to Aravamudan et al. ("Aravamudan"); filed October 7, 2005; published January 4, 2007; prior art under pre-AIA 35 U.S.C. § 102(e)
U.S. Patent Publication No. 2005/0283468 to Kamvar et al. ("Kamvar"); filed June 22, 2004; published December 22, 2005; prior art under pre-AIA 35 U.S.C. §§ 102(a), (b) and (e)

Grounds of Unpatentability	
1	Claims 1-11 are unpatentable as obvious over Sareen in view of Anick pursuant to 35 U.S.C. § 103.
2	Claims 12-14 are unpatentable as obvious over Sareen in view of Anick and Aravamudan pursuant to 35 U.S.C. § 103.
3	Claims 15-18 are unpatentable as obvious over Sareen in view of Anick, Aravamudan and Kamvar pursuant to 35 U.S.C. § 103.

III. Overview of the Subject Matter

A. Overview of the '317 Patent

The '317 patent discloses a computer database system and method for reconstructing partial search engine queries submitted by users at remote client

systems. The '317 patent purports to improve on existing search engine interfaces that “may be rigid and require users to submit full queries to perform searche[s],” by instead “allowing a flexible way for users to submit a partial query and reconstruct a full query based on the partial query.” '317 patent, Abstract. The '317 patent suggests performing a search using the reconstructed query. '317 Patent, Abstract.

Figure 2 is illustrative of the methods and systems of the '317 patent. According to Figure 2, full queries are determined (*i.e.*, “reconstructed”) based on (1) a partial query received from a user and (2) information stored in the computer database system about queries previously submitted by users. Search results for the reconstructed full queries may later be presented to the users alongside alternative reconstructed full queries.

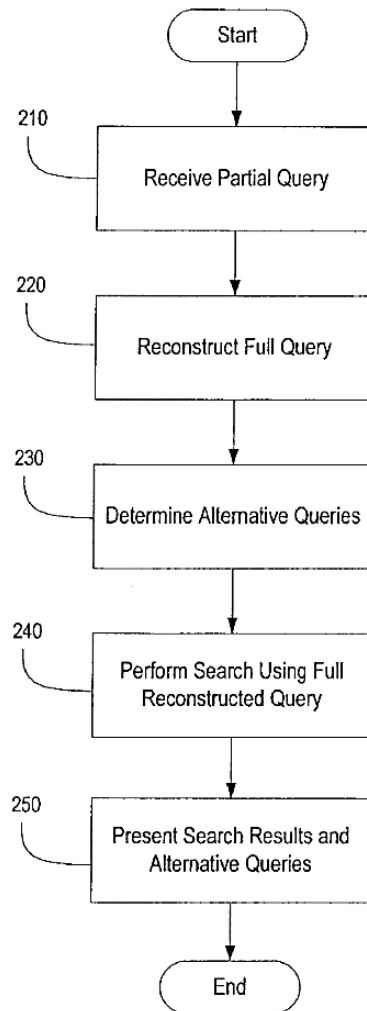


Figure 2

At step 210, a user submits a partial query, which is received by a server. '317 patent, 3:10-59. As explained by the applicant during the underlying prosecution, the user must commit a deliberate and conscious act of submission of the partial query before that partial query is processed (*see* Section III.B. below). The '317 patent describes the term “partial query” as “any abbreviated or incomplete search

query such that the submitted query is not fully representative of the entire search query desired by the user.” *Id.* at 3:12-15.

At step 220, a full query is then reconstructed from the partial query. For instance, the ’317 patent suggests that “auto insurance” or “American Idol™” could be reconstructed from the partial query “a i.” *Id.* at 3:10-59. The ’317 patent states that the determination of the full query may be based on previously submitted queries and/or indexes of previously submitted queries that were received by a server of the computer database system. *Id.* at 3:27-40, 4:14-6:13. At step 230, alternative reconstructed queries may also be determined, such as alternative full queries “baseball” and “basketball” when partial query “b” is submitted. *Id.* at 3:10-59. A search is then performed using the reconstructed full query (step 240) and the results of the search are presented to the user (step 250). *Id.*

Claim 1 of the ’317 Patent recites a computer database system for providing search results, the operation of which generally follows the sequence of Figure 2. The remaining independent claims 8 and 15 recite methods of query reconstruction that also generally follow the sequence of Figure 2.

B. Overview of the Prosecution History

The prosecution history underlying the ’317 patent and its parent, U.S. Patent No. 7,718,201, included various substantive rejections. EX1003, ¶¶44-50. Most relevant here, during the prosecution of the ’201 Patent, the examiner rejected the

then-pending application as obvious over U.S. Patent Application Publication No. 2006/0106769 to Gibbs (“Gibbs”) in view of Kamvar. *See* EX1005, 96-101. In amending the claims to overcome this rejection, the applicant argued that the prior art received and processed queries dynamically as the user typed the query, rather than upon a conscious and deliberate act of submission as required by the claims. For example, the applicant argued that the claimed “act of submitting the search query is a conscious and deliberate act and is different from the Gibbs situation of a user entering text and a software routine guessing at the search terms the user is entering.” EX1005, 114-115. When the rejection became final, the applicant again amended the claims and reiterated its alleged distinctions. The applicant argued that Gibbs and Kamvar disclose “detect[ing] a query as it is being entered by the user, guess[ing] at a set of predicted queries, and interrupt[ing] the user by presenting the predicted queries to the user,” whereas the claims require “a conscious and deliberate act by the user.” EX1005, 145-146.

C. The Level of Ordinary Skill in the Art

Based on the relevant factors, *see In re GPAC Inc.*, 57 F.3d 1573, 1579 (Fed. Cir. 1995), a person of ordinary skill in the art (“POSITA”) pertinent to the ’317 patent as of its earliest priority date would have had at least a four-year degree in computer science, computer engineering, or a closely related field and at least one year of experience in researching, designing, or developing computerized search

systems such as search engines. EX1003, ¶35. Additional formal education could substitute for professional experience, and significant professional experience could substitute for formal education. *Id.*

D. Claim Construction

In the district court dispute between Petitioner American Airlines, Inc. (“American Airlines”) and Patent Owner, the court construed several ’317 patent terms as follows:

Claim Term	Claim(s)	District Court’s Construction
“submitted”	1, 2, 4, 5, 8, 11–14	“submitted by a conscious and deliberate act of submission (for example by entering a carriage return or by pressing ‘enter’ or ‘send’) rather than just by entering text”
“query data management server”	4, 5	plain and ordinary meaning
“storing in memory data about previously received queries at the computer database system”	8	storing, in memory, data about previously submitted queries that have been received at the computer database system”
“better representative of the entire search query desired by the user”	1, 8	plain and ordinary meaning
“not fully representative of an entire search query desired by the user”	1, 8	plain and ordinary meaning

EX1011, 48-63.¹

The Petition shows that the '317 patent claims are invalid under the district court's constructions. The Board need not expressly construe these, or any other terms of the '317 patent, however, because the Petition shows that the '317 patent claims are invalid under any plausible construction of the terms.

IV. Overview of the Prior Art

A. Background and State of the Art

At the time the '317 patent was filed, search engines and their functionalities were well known to a skilled artisan. In fact, all of the concepts disclosed in the '317 patent were known well before the patent's priority date. For example, several references (cited in this Petition or during prosecution) evidence that Yahoo!'s competitors, including Microsoft and Google, had already disclosed and implemented search engines that reconstructed full queries from users' partial queries. One known implementation would enable the user to type out a partial query then hit enter or press a "search" button on the webpage to submit the query to a server across a network, including the Internet. For instance, a 2003 article by Peter G. Anick (the first inventor in the Anick reference) shows that the AltaVista search engine provided users with reconstructed full queries, denoted by "feedback

¹ In the District Court dispute between Petitioner Hilton Domestic Operating Company, Inc. ("Hilton") and Patent Owner (the "Hilton Litigation"), Hilton and Patent Owner have not yet engaged in claim construction.

terms” (e.g., “election returns” and “election results”) based on the submission of a partial query (e.g., “elections”).

Figure 1. Top of AltaVista search results page, showing feedback terms for the search “elections”.

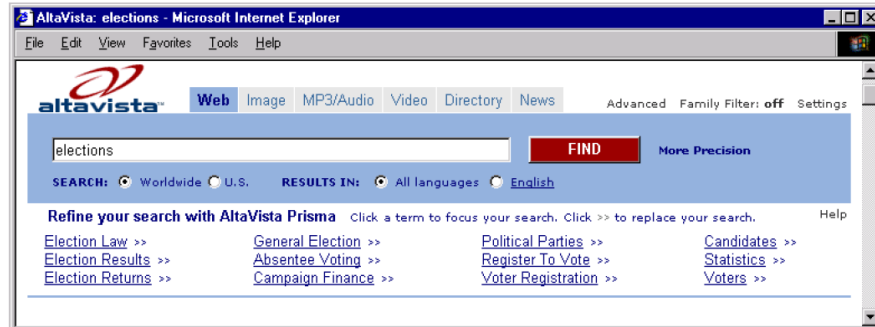
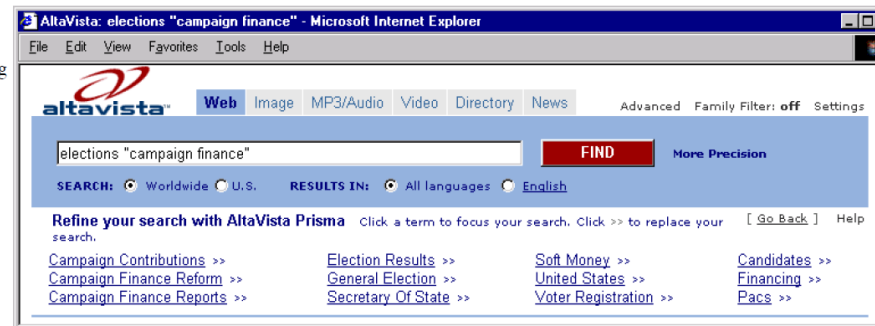
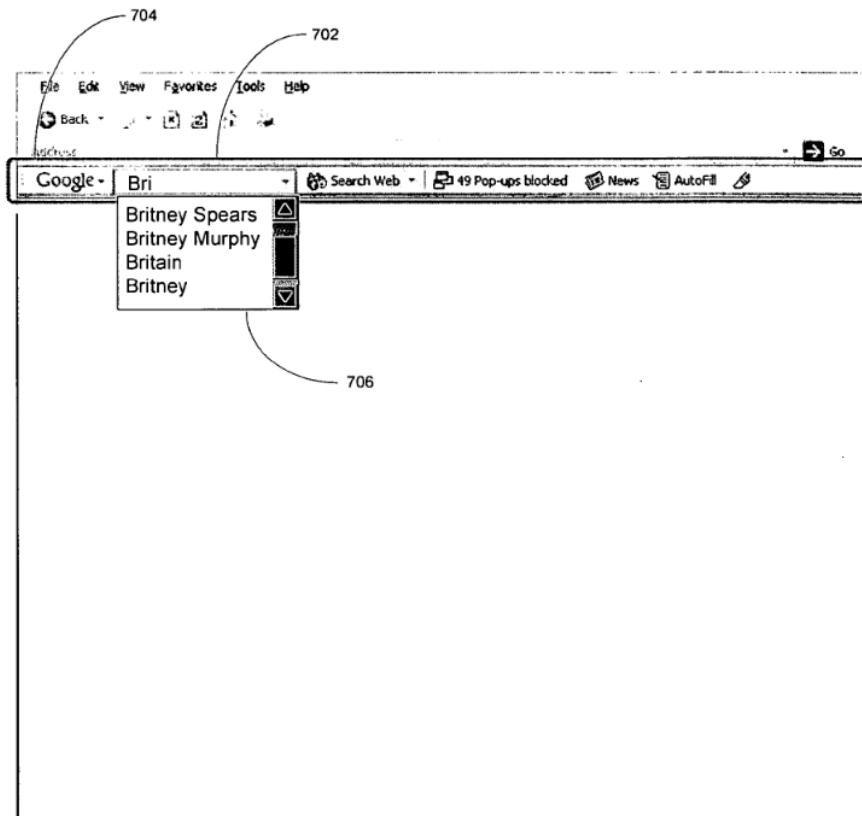


Figure 2. Top of AltaVista search results page, showing the search box and feedback terms presented after the user has clicked on the feedback term “campaign finance”.



EX1012, 89.

Under another well-known approach, the server would obtain partial queries as the user entered keystrokes in a search field (*see* Kamvar, Figure 7) and provide a dropdown list of possible reconstructed queries (e.g., reconstructing “Bri” as “Britain” or “Britney” in the example below).



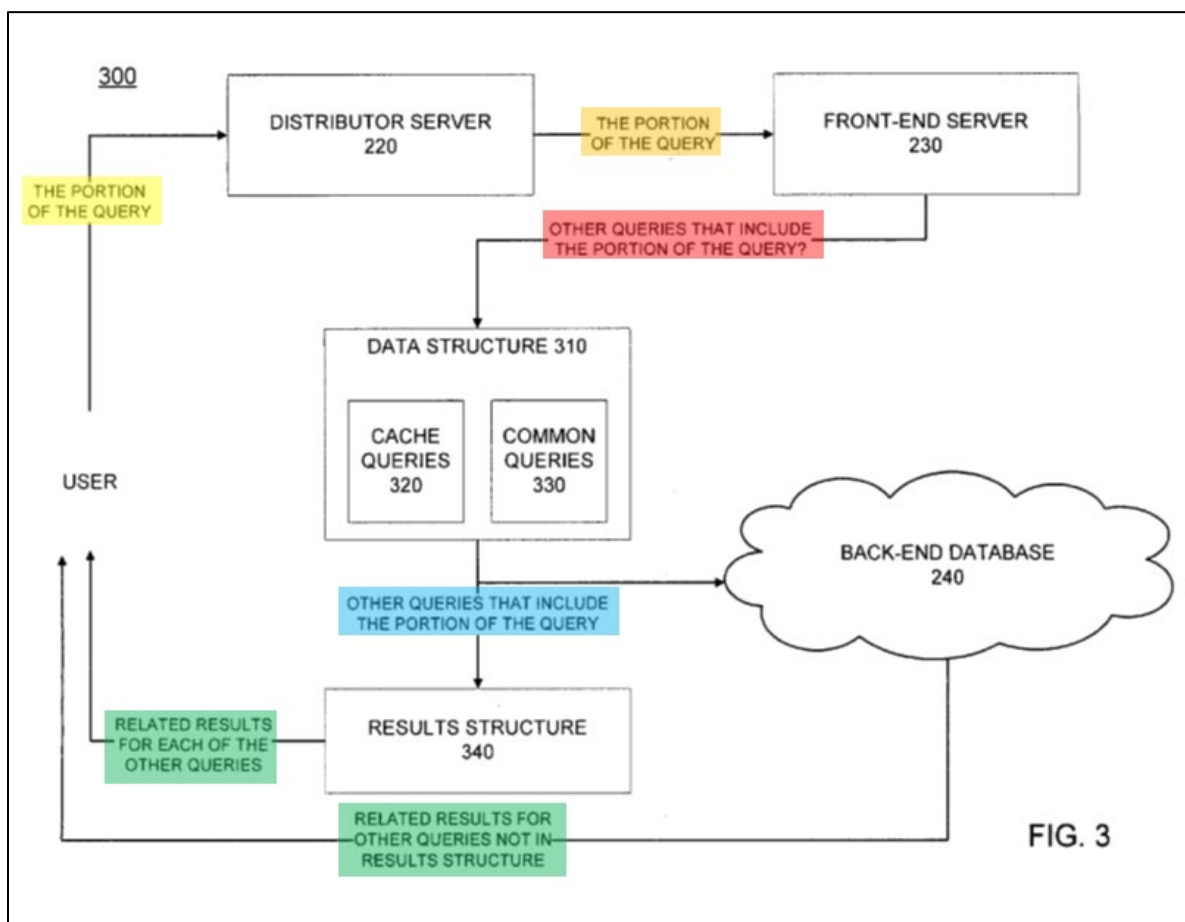
Kamvar, Figure 7. In either case, the search engine would match full queries to the user's partial query and perform a search using a selected reconstructed full query.

B. Overview of Sareen

Sareen, which was not before the Examiner during prosecution of the '317 patent, discloses "a system and method for obtaining related results for a portion of a query and for generating an updated set of queries for a cache of a server [...]" Other queries beginning with the portion of the query may be identified and obtained from a data structure that includes a server cache and a set of common queries. Once

the other queries are identified, results for the other queries are obtained from the server cache or from a back-end database.” Sareen, Abstract.

Sareen discloses various techniques to (1) determine “updated” full queries based on the partial query entered by the user, (2) provide the updated queries to the user, and (3) run searches using the updated queries. Figure 3 illustrates a query system implementing the techniques disclosed in Sareen.



Sareen, Fig. 3 (annotated).

In an exemplary embodiment, Sareen discloses that a user submits a portion of a query via the internet (highlighted yellow). A distributor server receives that

portion of a query and distributes it to one or more front-end servers. Sareen, [0029]-[0030]. Once the front-end server receives the portion of the query (highlighted orange), it searches for “other queries” that include the partial query entered by the user (highlighted red). Sareen, [0031]. To do so, the front-end server consults a data structure, in communication with local storage, to determine whether the local storage contains any recent queries or common queries that begin with the partial query entered by the user. Sareen, [0031]-[0036].

Sareen discloses that the front-end server may also consult a results structure that stores search results for the other queries obtained from the data structure (highlighted blue). The results structure may be part of the front-end server or may be housed in a separate component. Sareen, [0034]. When the front-end server does not find those results in the results structure, the front-end server may consult a back-end database, which contains stored search results and an index of web pages. Sareen, [0028], [0035].

C. Overview of Anick

Anick, which was not before the Examiner during prosecution of the '317 patent, discloses techniques for refining search engine queries submitted by a user. Figure 2 illustrates Anick's techniques, depicting a search engine browser displaying on a webpage search refinement suggestions (*e.g.* “Space Shuttle Challenger”) in response to a partial query entered by a user (*e.g.* “space shuttle”).

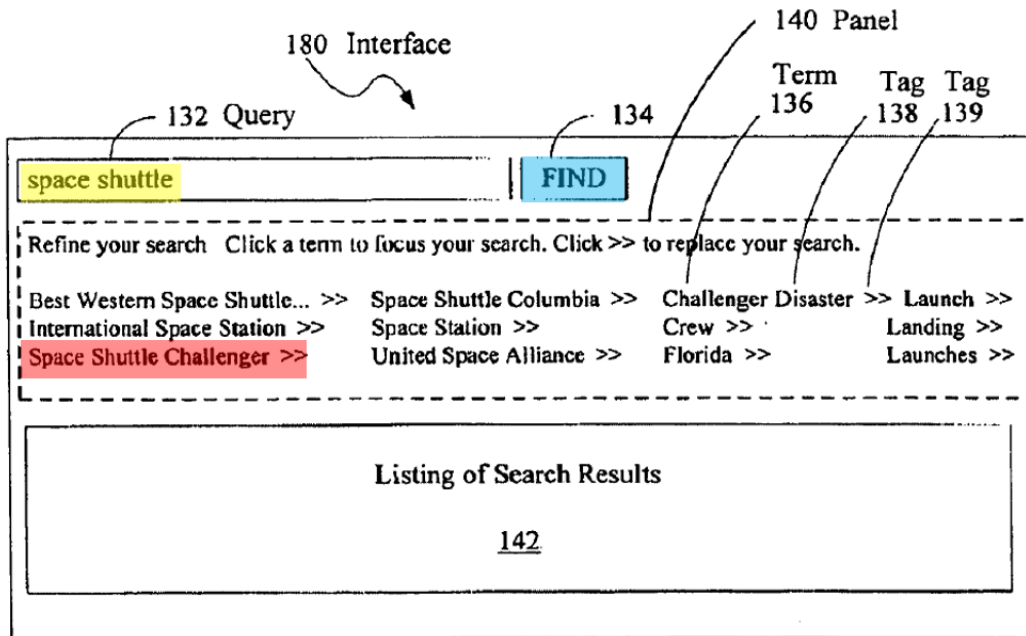


Fig. 2

Anick, Fig. 2 (annotated).

Anick discloses a user entering a partial query 132 at a client computer and submitting it to a search engine server by pressing the find button 134. Anick, 7:63-64. The server then processes the received query to provide both search results 142 and a set of candidate terms 136, each term being a search refinement suggestion that can improve the initial query. *Id.*, 7:65-8:7. The candidate terms are presented with two types of selectable tags shown as 138 and 139, which the user may click to append the term to the initial query or to replace the initial query with the candidate term. Anick, 8:10-16. Anick discloses that the addition or substitution of the

refinement term “provides the user with a query that quite possibly more closely matches the interests of the user.” Anick, 8:40-44.

Anick further discloses architectures for a search engine server, as illustrated by Figure 3.

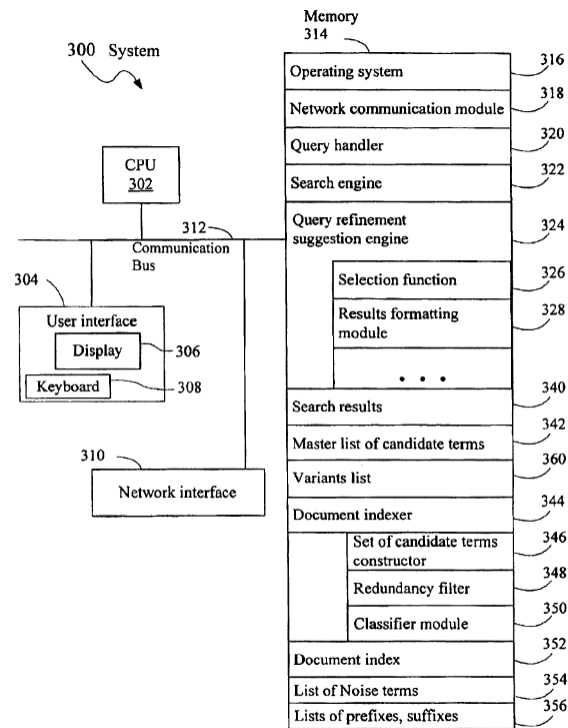


Fig. 3

The server includes: a query handler 320 for receiving a query from a client computer 100; a search engine 322 that searches a document index 352 for documents related to a query; and a query refinement suggestion engine 324. Anick, 9:10-41. The query refinement suggestion engine 324 comprises a selection function 326, which identifies candidate terms for query refinement, and a results formatting module for presenting the candidate terms. Anick, 9:10-41.

D. Overview of Aravamudan

Aravamudan discloses methods and systems for processing search queries that are entered as acronyms or prefixes delimited by spaces or other separators. Aravamudan, [0010]. One of Aravamudan's disclosed search processes is illustrated in Figure 4.

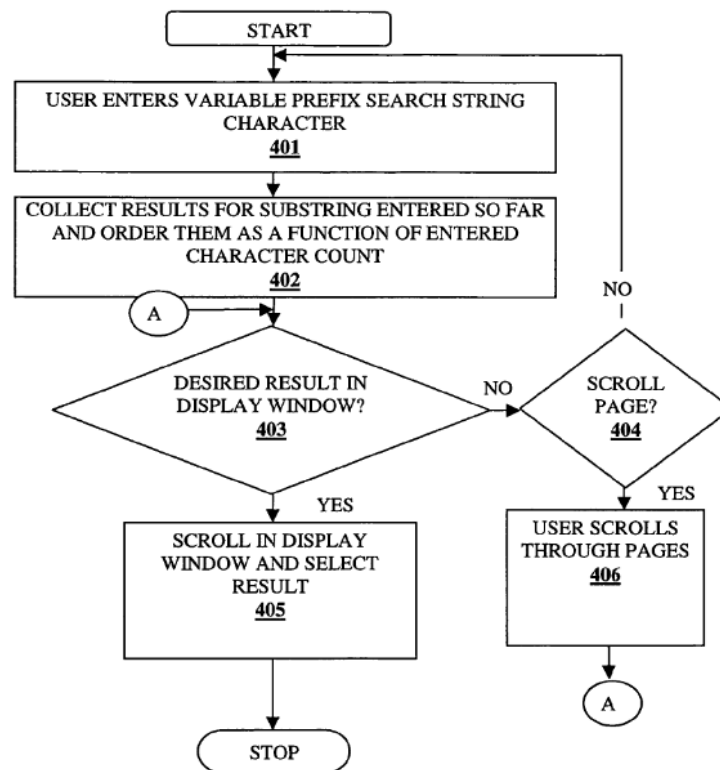


FIG. 4

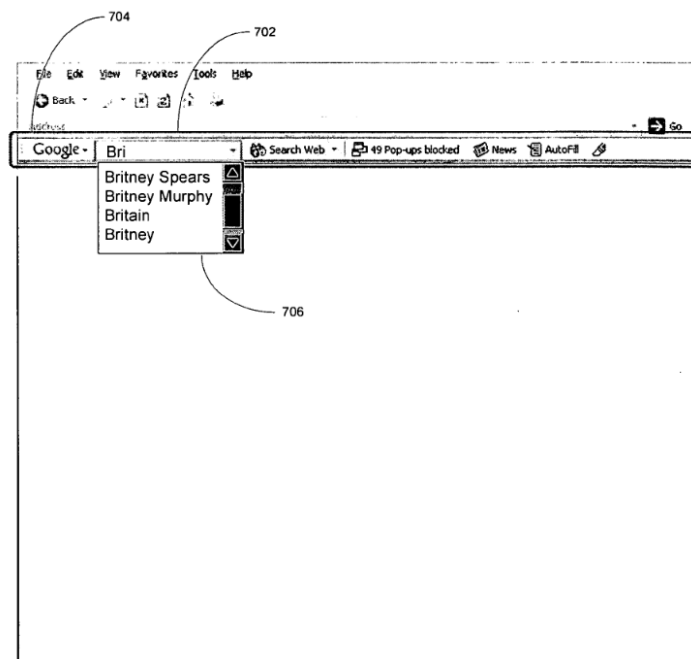
A user may start the process by entering a search string comprising prefixes for the “intended query at 401. For example, to represent ‘Brad Pitt,’ the user may enter B P, BR P, B PI etc.” Aravamudan, [0033]. A server system receives that search string (e.g., “B P”) and retrieves search results by matching each delimited substring to

relevant terms that may form the intended queries. *Id.* The system searches for the results of the matched terms and presents the results in a display window 403 by order of relevance. *Id.* The user may then select “the desired result” from the display (according to steps 403 and 405) or scroll through the pages if the desired result does not appear in the display (under steps 404 and 406). *Id.*

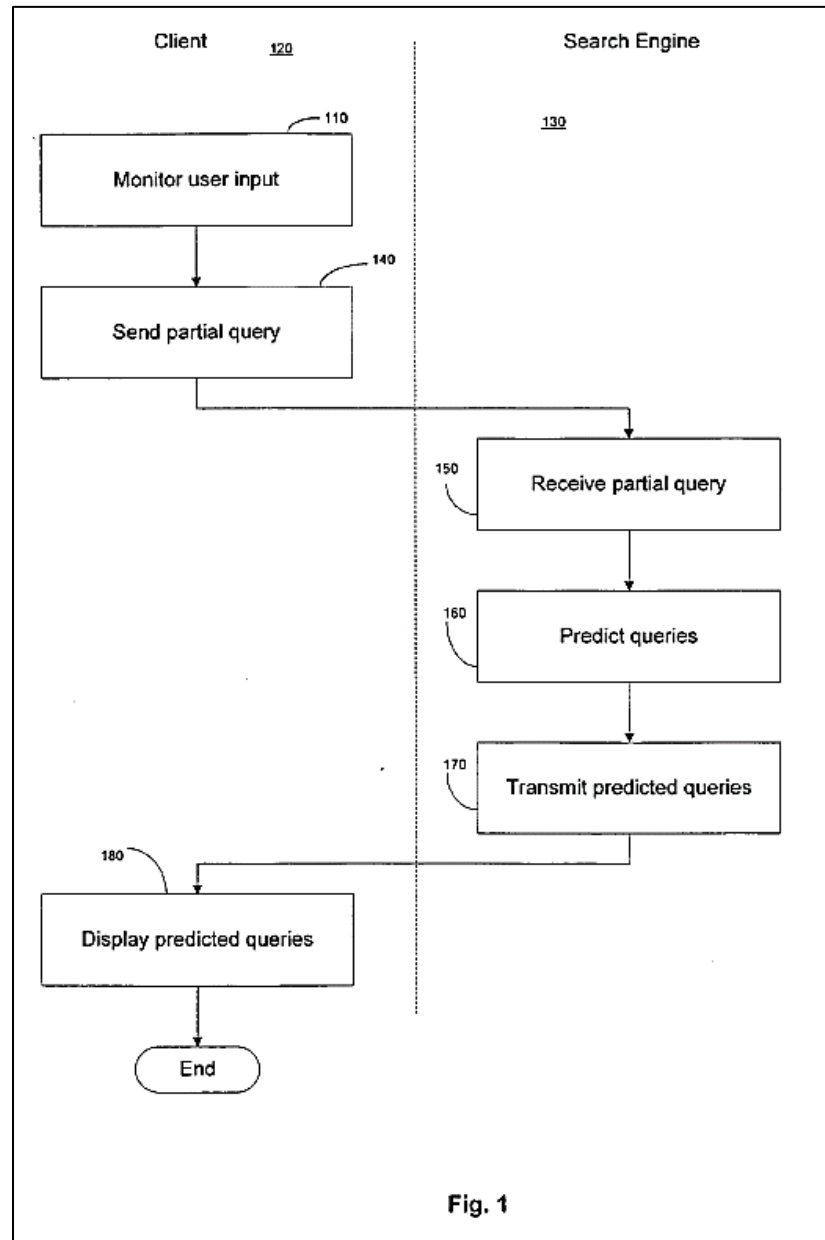
E. Overview of Kamvar

Kamvar discloses methods and systems for monitoring a user’s input of a search query, sending the input as a partial query to a server and determining, at the server, a set of predicted queries based on the partial query. Kamvar, [0005]. Figure 7 provides an illustration of a search engine providing reconstructed search queries determined from a partial query.

Patent Application Publication Dec. 22, 2005 Sheet 7 of 11 US 2005/0283468 A1



An exemplary process for the query prediction is depicted in Figure 1:



Kamvar discloses that partial queries may be matched to predicted queries using a dictionary listing partial terms and corresponding popularity values. The dictionary may be created from previously submitted search queries and may track the popularity over time of each term portion, assigning to each term a popularity value

“that indicates how popular the associated term portion may be at any given movement in time.” Kamvar, [0039]-[0042].

V. Ground 1: Claims 1-11 are Obvious in view of Sareen and Anick

A. Claim 1

1. Preamble

The preamble of claim 1 recites “[a] computer database system for providing search results to a user in response to user submissions over a data network.” To the extent the preamble is limiting, the combination of Sareen and Anick renders this limitation obvious. EX1003, ¶¶74-77.

Sareen discloses “a system and method for obtaining results for a portion of a query and for generating an updated set of queries.” Sareen, Abstract. Sareen further discloses that its system “may be utilized to provide search results for queries, such as queries or portions of queries entered in an internet search engine through, for example, the computer.” Sareen, [0025]. A skilled artisan would understand these results to be returned in response to “*user submissions over a data network.*” EX1003, ¶¶76-77.

Sareen discloses that its system may comprise “a distribution server 220 for receiving a portion of a query,” “a front-end server 230,” “a data structure 310,” including local storage, a “results structure 340” and a “back-end database 240.” Sareen, [0029]-[0031], [0034]-[0035]; *see also id.*, Fig. 3 (illustrating an exemplary

system). Sareen further discloses that its computer system “may operate in a network environment using logical connections to one or more remote computers,” such as a remote computer used to submit the user queries. Sareen, [0022]-[0023]. A skilled artisan would understand Sareen’s system to be a “*computer database system*.” EX1003, ¶75.

2. Database

Claim 1 recites “*a database configured to store information about events in the computer database system*.” The combination of Sareen and Anick renders this limitation obvious. EX1003, ¶¶78-83.

Sareen discloses “local storage” that “may contain queries and their corresponding results.” Sareen, [0027]. Sareen discloses that the stored data “may be a set of recent queries and results, such as those from a predetermined time period.” Sareen, [0027], [0034]. Sareen discloses that the local storage may also “include a set of common queries,” which are stored “for use in providing queries that have the same portion as the portion of the search query.” Sareen, [0032], [0037]. A skilled artisan would understand Sareen’s disclosure of storing this information in local storage to teach a “*database configured to store information*.” EX1003, ¶¶79-80.

Patent Owner correctly conceded in its claim construction briefing in the American Airlines Litigation that a database configured to store previously

submitted queries is sufficient to satisfy this claim limitation. EX1013, 26. A skilled artisan would understand Sareen's "local storage," which stores previously submitted queries and corresponding query search results, to disclose the claimed "*database configured to store information about events in the computer database system.*" EX1003, ¶80.

To the extent Sareen does not expressly disclose "*a database configured to store information about events in the computer database system,*" it would have been obvious to modify Sareen's system to include one. EX1003, ¶81. Sareen already discloses storing information about previously submitted queries in local storage. Sareen, [0027], [0034]. Sareen further discloses flexibility in both the location and the persistence of that storage. For example, Sareen discloses that the local storage "may be part of the front-end server 230" or it could be part of the data structure 310, which may be "its own component in the system." Sareen, [0027], [0031]. As to persistence, Sareen discloses using three hours as an example for the recent queries and one week as the duration for the common queries. Sareen, [0027], [0037]-[0038].

A skilled artisan would have been motivated to modify Sareen to include a "database" to further increase the persistence of the system and achieve the corresponding benefits. EX1003, ¶82. Moreover, a skilled artisan would understand that the storage in Sareen could be modified to achieve persistence without

sacrificing the speed of accessing this information locally rather than the back-end database. EX1003, ¶82. A skilled artisan would have had a reasonable expectation of success in making this modification because storing the information in a database, as opposed to Sareen’s local storage, would have been a routine design choice within the ordinary abilities of a skilled artisan. EX1003, ¶83.

3. Query Reconstruction Server

Claim 1 recites “*a query reconstruction server in data communication with the database and operative to receive a partial query submitted at a remote user client system by a user seeking search results matching the submitted partial query and, in response to the received partial query, determine a full query based on: (i) the received partial query; and (ii) information stored in the database about queries previously-submitted by users.*” The combination of Sareen and Anick renders this limitation obvious. EX1003, ¶¶84-104.

a. Data Communication with the Database

Sareen discloses “*a query reconstruction server in data communication with the database*” as recited in Claim 1. EX1003, ¶¶85-88. For example, Sareen discloses a “front-end server 230” that “receives [a] portion of the query” entered by the user and that “may be responsible for performing a search in order to obtain other queries that have the same portion of the query” (*i.e.*, reconstructing a full query). Sareen, [0030]; *see also id.*, [0005]. A skilled artisan would understand Sareen’s

“front-end server” to be an example of a “*query reconstruction server*” recited in Claim 1. EX1003, ¶86.

Sareen further discloses that the front-end server communicates with the local storage (*i.e.*, “*database*”) to obtain these other queries. Sareen, [0031] (“In order to obtain these queries, the front-end server 230 may consult a data structure 310. . . . The data structure 310 may include a server cache, such as cache queries 320.”), [0032] (“The data structure 310 may also include a set of common queries.”), [0034]. A skilled artisan would understand that Sareen’s disclosure of the front-end server consulting with local storage teaches “*data communication with the database.*” EX1003, ¶87. Alternatively, a skilled artisan would understand that, as modified to include a separate database storing information about previously submitted queries (*see* Section VA.2), Sareen’s computer database system would include a front-end server in data communication with a database. EX1003, ¶88.

b. Operative to Receive a Partial Query Submitted by a User

Sareen and Anick render obvious a front-end server “*operative to receive a partial query submitted at a remote user client system by a user seeking search results matching the submitted partial query*” as recited in Claim 1. EX1003, ¶¶89-100.

Sareen discloses a “distributor server 220 for receiving a portion of a query,” such as “the first three characters of a query” (*i.e.*, “*a partial query*”). Sareen, [0029]; *see also* Sareen, [0003] (contrasting conventional systems requiring a user to submit a full search term with the disclosed system processing partial queries submitted by a user). Sareen discloses that the “front-end server 230 receives the portion of the query” from the distribution server. Sareen, [0030]. A skilled artisan would understand that Sareen’s front-end server (*i.e.*, the “*query reconstruction server*”) therefore, is “*operative to receive a partial query.*” EX1003, ¶90.

A skilled artisan would likewise understand that the partial query received by the front-end server was “*submitted at a remote user client system by a user seeking search results matching the submitted partial query.*” EX1003, ¶¶91-92. Sareen discloses that a “user, who desires results to the query, may type ***or enter*** a portion of the query in an internet search engine” through a remote computer. Sareen, [0029] (emphasis added); *see also id.*, [0025] (“A user may enter commands and information into the computer 110 through input devices such as a keyboard 162.”); *see also* Sareen, [0003]. Sareen likewise discloses that a query or partial query “originates at the internet 210, where a user may insert a query for which they desire to obtain relevant results, such as web pages directed to the query.” Sareen, [0025].

To the extent Sareen does not expressly disclose that the query entered by a user and received by the front-end server has been “submitted” as that term has been

construed by the district court in the American Airlines Litigation, it would have been obvious to a skilled artisan to modify Sareen to include this functionality in view of at least the disclosures of Anick. EX1003, ¶¶93-100; EX1011, 53 (construing “submitted” as “submitted by a conscious and deliberate act of submission (for example by entering a carriage return or by pressing ‘enter’ or ‘send’) rather than just by entering text.”).

Anick discloses processing a search query that has been “submitted by a client computer 100 to a search engine server.” Anick, 7:43-44. For example, Anick discloses that “when [the] find button 134 is pressed, query 132 is sent from client computer 100 to search engine server 110.” Anick, 7:63-64. Anick discloses that this query is then received and processed by the search engine. Anick, 7:65-67 (“Upon receiving query 132, search engine server 110 processes received query 132 and sends search results and search refinement suggestions back to client computer 100.”). A skilled artisan would understand Anick’s disclosure of a partial query “submitted” by a client by pressing the find button to teach a “*partial query submitted*” at a remote user client under the district court’s construction. EX1003, ¶94.

A skilled artisan would have been motivated to combine with a reasonable expectation of success Sareen’s computer database system operative to receive and process a partial query entered by a user with Anick’s teaching of receiving and

processing a partial query submitted by a user. EX1003, ¶¶95-100. A skilled artisan would have been motivated to make this combination to comport with a user's expectation in using a search engine and to achieve efficiency benefits associated with delaying processing until the user consciously submits the query. EX1003, ¶¶95-96.

As Dr. Bederson explains, a skilled artisan would understand that a user entering a search term in a search engine would not expect a conscious and deliberate act of submission (such as hitting enter) after typing a partial query to meaningfully alter the functionality of the system. EX1003, ¶96. Thus, to the extent Sareen does not expressly disclose receiving and processing a partial query that has been submitted through a deliberate act of a submission, a skilled artisan would have been motivated to modify Sareen to process queries that have been consciously submitted to comport with the user's expectation in committing a deliberate or conscious act of submission. EX1003, ¶¶95-96.

Alternatively, a skilled artisan would have been motivated to modify Sareen to process only queries that have been deliberately submitted, as opposed to processing queries dynamically as the user types. EX1003, ¶¶97-100. A skilled artisan would have been motivated to make this modification to use bandwidth more efficiently. EX1003, ¶¶97-100. Sareen's disclosures focus on resource conservation, discussing the use of the front-end cache to preserve resources.

Sareen, [0001] (“[I]t is more efficient and less time-consuming to consult the front-end cache.”). Specifically, Sareen states that its “portion-querying mechanism would be further desirable if resources were preserved and operations completed faster. . . .” Sareen, [0004]. Modifying Sareen to process queries only upon a deliberate act of submission, as opposed to dynamically processing queries as the user types, would have been a natural extension of Sareen’s desire to preserve resources such as reducing bandwidth by avoiding having to communicate with the back-end server. EX1003, ¶¶97-98. This is particularly true in view of Sareen’s disclosure that the user does not always want to search the first few characters of its queries (*i.e.*, “www,” “http”). *See* Sareen, [0039]. A skilled artisan therefore, would have been motivated to modify Sareen to wait for the user to consciously and deliberately submit the query prior to processing the query to determine a fully query, as Anick discloses. EX1003, ¶98.

Sareen mentions that it would be “desirable for the results to automatically adjust” as a user continues to type. Sareen, [0003]. A skilled artisan would not have understood this disclosure to teach away from combining Anick and Sareen. EX1003, ¶99. A skilled artisan instead would have understood this disclosure to indicate a routine design choice between processing queries in real-time and waiting until a user consciously submits the query, involving an evaluation of the trade-offs. EX1003, ¶99. On one hand, Sareen mentions there are benefits to processing a

partial query as the user types. Sareen, [0003]. On the other hand, however, a skilled artisan would recognize that there would be efficiency benefits in waiting to process the query until the user commits a deliberate and conscious act of submission. EX1003, ¶99 (citing EX1014, 13:60-63 (“To conserve processing resources or for other reasons, a limit can be set that provides a maximum number of times that a particular query can be modified or otherwise refines an update algorithm.”)). Each choice has relative advantages and tradeoffs, and one skilled in the art would have recognized them and weighed them in choosing how to implement the combined system. EX1003, ¶99. This type of choice was common when designing a system, and one skilled in the art would have routinely made such decisions. EX1003, ¶99.

A skilled artisan would have had a reasonable expectation of success in making this combination because waiting to process a query until it has been deliberately submitted by a user was a routine technique at the time the ’317 patent was filed, *see e.g.*, Anick at 7:43-44. EX1003, ¶100. A skilled artisan would have found it within his or her ordinary abilities to modify Sareen’s system to incorporate Anick’s disclosure of processing submitted queries. EX1003, ¶100.

c. Operative to Determine a Full Query

Sareen discloses that “*in response to the received partial query,*” the front-end server will “*determine a full query based on: (i) the received partial query; and*

(ii) information stored in the database about queries previously-submitted by users”
as recited in Claim 1. EX1003, ¶¶101-104.

Sareen discloses determining a full query by comparing the received partial query to previously submitted queries. Specifically, Sareen explains that “[o]nce the front-end server 230 receives the portion of the query, the front-end server 230 may be responsible for performing a search in order to obtain other queries that have the same portion as the portion of the query.” Sareen, [0031]. A skilled artisan would understand Sareen’s disclosure of “obtain[ing] other queries that have the same portion as the portion of the query” to mean that Sareen’s front-end server “*determine[s] a full query.*” EX1003, ¶102.

Sareen discloses that in determining full queries, the front-end server “may consult a data structure 310.” Sareen, [0031]. The data structure in Sareen may store in local storage (i.e., the “*database*”) both “the most recent queries” submitted by a user and a set of “frequently requested queries.” Sareen, [0031]-[0032], [0038]; EX1003, ¶¶103-104. In consulting the stored information, the front-end server compares the portion of the query (i.e., “*the received partial query*”) to recent queries and frequently requested queries (i.e., “*queries previously submitted by users*”), thereby “*determin[ing] a full query based on: (i) the received partial query; and (ii) information stored in the database about queries previously submitted by users.*” EX1003, ¶103.

As discussed above, to the extent the local storage in Sareen is not a “*database*,” it would have been obvious to modify Sareen to instead store the information about previously submitted queries in a database. *See* Section V.A.2.² A skilled artisan would understand Sareen to determine the full query in the same way with respect to the database as it did with respect to the local storage, *i.e.*, it would consult the information about previously submitted queries that is stored in the database. EX1003, ¶104. Thus, a skilled artisan would have understood Sareen and Anick, as modified to include a database, to also render this limitation obvious. EX1003, ¶104.

4. Full Query Better Representative of Entire Query

Claim 1 further recites “*wherein the submitted partial query comprises an abbreviated or incomplete search query which is not fully representative of an entire query desired by the user and the full query is better representative of the entire search query desired by the user.*” The combination of Sareen and Anick renders this limitation obvious. EX1003, ¶¶105-107.

Sareen discloses a “*partial query*” that is a “portion of a query, such as the first few characters” (*i.e.*, “*an abbreviated or incomplete search*”). Sareen, [0003]. Sareen distinguishes the entered “portion of a query” from a “full search term or

² As used herein, references to previous discussions should be understood to incorporate those discussions in full.

query.” Sareen, [0003]. For example, Sareen discloses a “portion of a query” entered by a “a user, who, desiring results related to the query, enters the portion of the query in an internet search engine for example.” Sareen, [0039]. A skilled artisan therefore, would understand that Sareen’s portion of a query (e.g., the first few characters of a query) is “*not fully representative of an entire query desired by the user.*” EX1003, ¶106.

Sareen contrasts its systems with previous systems where a user’s submission of a partial query often returned an error message or incorrect results. Sareen, [0003]. Sareen’s system instead returns relevant results for a portion of a query “by identifying other queries beginning with the portion of the query and obtaining the results of the other queries.” Sareen, [0005]. A skilled artisan therefore, would understand that Sareen’s “other queries beginning with the portion of the query” (i.e., “*full quer[ies]*”) are “*better representative of the full entire search query desired by the user.*” EX1003, ¶107.

B. Claim 2

Claim 2 recites “[t]he computer database system of claim 1 wherein the query reconstruction server is further operative to determine as the full query the most frequently submitted, previously-submitted query that matches partial words of the partial query.” The combination of Sareen and Anick renders this limitation obvious. EX1003, ¶¶108-109.

As discussed above, Sareen discloses that the front-end server consults local storage containing both “the most recent queries” submitted by a user and a set of “frequently requested queries.” Sareen, [0031]-[0032], [0038]. Sareen further discloses that the stored information may “be a list of a predetermined number of most frequently requested queries beginning with the portion of the query . . . sorted in a descending order of frequency, so that the most common requested query beginning with the portion of the query is first listed.” Sareen, [0033]. A skilled artisan therefore, would understand Sareen to teach determining as the full query the “most common requested query beginning with the portion of the query” (*i.e.*, “*most frequently-submitted, previously-submitted query that matches the partial query*”). EX1003, ¶109.

C. Claim 3

Claim 3 recites “[t]he computer database system of claim 2 further comprising: a web server operative to receive search results produced by a search engine and deliver web pages that include the received results to a browser of the user.” The combination of Sareen and Anick renders this limitation obvious. EX1003, ¶¶110-113.

Sareen discloses that its back-end database 240 may contain search results, including web pages. Sareen, [0028]. Sareen also discloses a results structure 340, which “contains results for queries” that have been generated by a search engine.

Sareen, [0034]. “Upon consultation with the results structure 340, results for queries that begin with the portion of the query are obtained and provided to the user.” Sareen, [0034]. The presented results “may be, for example, a multiple selection of web pages” (*i.e.*, “*web pages that include the received results*”). Sareen, [0034]; EX1003, ¶111.

A skilled artisan would understand that the results are “obtained and provided to the user” by a “*web server configured to receive . . . and deliver*” these search results. EX1003, ¶112. For example, Sareen discloses a “distributor server” (*i.e.*, a “*web server*”) which receives a user’s search from “a web search engine.” Sareen, [0001], [0025] (“The internet 210 may be connected to a distributor server 220, which receives the query from the internet.”). Sareen discloses that the distributor server transmits the query to a front-end server (also a “*web server*”), which then obtains full queries and the corresponding search results, including web pages. Sareen, [0031]-[0034]. As illustrated in Figure 2, the distributor server 210 serves as the gateway between the back-end database, front-end server, and the internet. Sareen, Fig. 2. A skilled artisan therefore, would understand that the results “which are obtained and provided to the user” are “*receive[d]*” and “*deliver[ed]*” by a front-end server 230 via the distributor server 220 (each being “*a web server*”). EX1003, ¶112.

Sareen discloses a user interacting with the computer database system through a remote computer, including a search engine. Sareen, [0025]; EX1003, ¶113. A skilled artisan would understand this interaction to involve a user entering search terms and receiving corresponding search results through a search engine browser. EX1003, ¶113. A skilled artisan therefore would have understood that the distributor server is configured to deliver the search results, including the disclosed webpages, to the user via the search engine browser on the user's personal computer (i.e., “*deliver . . . to a browser of the user*”). EX1003, ¶113.

D. Claim 4

Claim 4 recites “[t]he computer database system of claim 1 further comprising: a query data management server in data communication with the query reconstruction server and the database, the query data management server being operative to provide to the query reconstruction server the information stored in the database about queries previously-submitted by users.” The combination of Sareen and Anick renders this limitation obvious. EX1003, ¶¶114-120.

Sareen discloses a “*query data management server in data communication with the query reconstruction server and the database.*” EX1003, ¶115. For example, Sareen illustrates a data structure 310 in data communication with the front-end server (i.e., “*the query reconstruction server*”) and the local storage (i.e., “*the database*”). Alternatively, a skilled artisan would have understood that, as

modified to include a separate database storing information about previously submitted queries (*see* Section VA.2), Sareen’s computer database system would include the data structure 310 consulting (*i.e.*, being in “*data communication with*”) the database as opposed to the local storage (if deemed not to be a database) in Sareen. EX1003, ¶115. A skilled artisan would understand Sareen’s “data structure 310” to disclose the claimed “*query data management server.*” EX1003, ¶115.

As discussed above, *see* Section V.A.3.c, Sareen discloses that the front-end server “may consult a data structure 310” in determining the full queries. Sareen, [0031]. A skilled artisan would understand that the data structure in Sareen may be in data communication with local storage comprising “the most recent queries” submitted by a user and a set of “frequently requested queries.” EX1003, ¶116 (citing Sareen, [0031]-[0032], [0038]). When the front-end server consults the data structure, the data structure provides the front-end server (*i.e.*, “*the query reconstruction server*”) with recent queries and frequently requested queries (*i.e.*, “*information . . . about queries previously submitted by users*”). As discussed above, *see* Section V.A.3.c, a skilled artisan would understand that this information is “*stored in the database.*” EX1003, ¶116.

A skilled artisan would understand that Sareen’s system is flexible as to the location of the data structure. EX1003, ¶117. For example, Sareen discloses that the data structure 310 could be either a part of the front-end server 230 or a separate

component in the system. Sareen, [0031]; *see also id.* Fig. 3. If the data structure is a separate component, as illustrated in Figure 3 of Sareen, a skilled artisan would understand that it would be included in a separate server. EX1003, ¶117. A skilled artisan therefore would understand that Sareen’s data structure 310 could be on its own server (*i.e.*, “*query data management server*”). EX1003, ¶117. A skilled artisan would understand that, if the data structure were located on a separate server, the functionality of Sareen’s system otherwise would be unchanged. EX1003, ¶117.

To the extent the Board finds that Sareen does not expressly disclose a “*query data management server*” it would have been obvious to modify Sareen’s system to include a server to perform the functionality described with respect to Sareen’s data structure. EX1003, ¶¶118-119. A skilled artisan would have been motivated to make this modification with a reasonable expectation of success at least because of the disclosed flexibility in Sareen’s system. EX1003, ¶¶118. A skilled artisan would have understood that Sareen’s data structure could be implemented as part of a server, as expressly disclosed by Sareen. EX1003, ¶118 (citing Sareen, [0031] (The data structure “may be part of the front-end server 230.”)). A skilled artisan would have been motivated to implement the data structure as part of a server that is separate from the front-end server because separating services on different computers is a well-known way to implement a database system. Sareen, [0031] (The data structure “may be part of the front-end server 230.”); EX1003, ¶118.

Moreover, there are only a limited number of places that the data structure could be housed outside of the front-end server. EX1003, ¶¶118-119. One of those limited choices would be a separate server. EX1003, ¶¶118-119. Placing the data structure on a separate server would have been within the routine abilities of a skilled artisan. EX1003, ¶120.

E. Claim 5

Claim 5 recites “[t]he computer database system of claim 4 wherein the query data management server is further configured to: retrieve from the database historical query data relating to a plurality of previously submitted search queries; build a query index based on the historical query data; store the query index in the database; and provide the query index to the query reconstruction server.” The combination of Sareen and Anick renders these limitations obvious. EX1003, ¶¶121-124.

Sareen discloses that its data structure (“query data management server”) is in data communication with local storage comprising both “the most recent queries” submitted by a user and a set of “frequently requested queries” (i.e., “historical query data relating to a plurality of previously submitted queries”). Sareen, [0031]-[0032], [0038]; EX1003, ¶122. Sareen discloses that the data structure may use the historical query data to build a query index that is then stored in the local storage for fast retrieval. For example, Sareen discloses that the “data structure may be an

augmented trie structure, which is a structure that facilitates fast retrieval of information related to a query or a portion of query.” Sareen, [0033]. The trie structure (*i.e.*, “*query index*”) may include “possible portions of a query along with additional information.” Sareen, [0033]. “The information related to each portion of the query may for example be a list of a predetermined number of most frequently requested queries beginning with the portion of the query.” Sareen, [0033]. A skilled artisan therefore, would understand that the data structure is “*configured to . . . build a query index based on the historical query data.*” EX1003, ¶123.

When the front-end server consults the data structure, the front-end server (*i.e.*, “*the query reconstruction server*”) is provided with the information in the data structure, including for example, the list “of a predetermined number of most frequently requested queries beginning with the portion of the query.” Sareen, [0033]-[0034]. A skilled artisan therefore, would understanding that Sareen’s data structure is “*configured to provide the query index to the query reconstruction server.*” EX1003, ¶124.

F. Claim 6

Claim 6 recites “[t]he computer database system of claim 4 wherein the query data management server is further configured to: retrieve from the database information about queries previously submitted to the computer database; filter the retrieved information about the queries; and construct one or more indexes from the

filtered query information, the query data management server providing the constructed indexes to the query reconstruction server for use as the information stored in the database about queries previously-submitted by users when the query reconstruction server determines the full query.” The combination of Sareen and Anick renders these limitations obvious. EX1003, ¶¶125-128.

As discussed above with respect to Claim 5, a skilled artisan would understand that Sareen discloses a “*query data management server [] configured to: retrieve from the database information about queries previously submitted to the computer database; . . . construct one or more indexes . . . [and] provid[e] the constructed indexes to the query reconstruction server for use as the information stored in the database about queries previously-submitted by users when the query reconstruction server determines the full query.”* See Section V.E *supra*; EX1003, ¶126.

A skilled artisan would further understand that Sareen discloses “*filter[ing] the retrieved information about the queries.”* EX1003, ¶127. As discussed above, Sareen discloses that its data structure (*i.e.*, “*query data management server*”) may be an augmented trie structure, which a skilled artisan would understand to be an existing data structure that has been augmented (*e.g.*, “*filter[ed]*”) to fit an application’s needs. EX1003, ¶127. In Sareen, this “*filter[ing]*” may be in the form of modifying the historical data to form a list “of a predetermined number of most frequently requested queries beginning with the portion of the query.” Sareen,

[0033]. A skilled artisan would understand that the data structure’s “trie structure” (*i.e.*, “*query index*”) is constructed using the filtered list of most frequently requested queries beginning with the portion of the query (*i.e.*, “*the query data management server is configured to . . . construct one or more indexes from the filtered query information*”). EX1003, ¶128 (citing Sareen, [0033]).

G. Claim 7

Claim 7 recites “[t]he computer database system of claim 6 wherein the query data management server is configured to select, as the filtered query information, most popular queries from the retrieved information about queries previously submitted.” The combination of Sareen and Anick renders this limitation obvious. EX1003, ¶¶129-130.

As discussed above with respect to Claim 6, Sareen discloses filtering the historical query data to form a list “of a predetermined number of most frequently requested queries beginning with the portion of the query.” Sareen, [0033]; *see* Section V.F. A skilled artisan would understand the “most frequently requested queries” to be the “*most popular queries from the retrieved information about queries previously submitted.*” EX1003, ¶130.

H. Claim 8

1. Preamble

The preamble of claim 8 recites “[a] *computer implemented method for a computer database system.*” To the extent the preamble is limiting, the combination of Sareen and Anick renders this limitation obvious. EX1003, ¶¶131-133.

Sareen discloses “a system and method for obtaining results for a portion of a query and for generating an updated set of queries for a cache of a server.” Sareen, Abstract. A skilled artisan would understand that the disclosed method is “*for a computer database system.*” EX1003, ¶133. For example, as discussed above with respect to Claim 1, Sareen discloses an exemplary “*computer database system*” comprising “a distribution server 220 for receiving a portion of a query,” “a front-end server 230,” “a data structure 310,” a “results structure 340” and a “back-end database 240.” Sareen, [0029]-[0031], [0034]-[0035]; *see also id.*, Fig. 3 (illustrating an exemplary system); EX1003, ¶132. Sareen likewise discloses that its method may be implemented on its computer database system “to provide search results for queries, such as queries or portions of queries entered in an internet search engine through, for example, the computer 110.” Sareen, [0025]; EX1003, ¶133.

2. Storing in Memory

Claim 8 recites “*storing in memory data about previously received queries at the computer database system.*” The combination of Sareen and Anick renders this limitation obvious. EX1003, ¶¶134-135.

As discussed above with respect to Claim 1, Sareen discloses storing in “local storage” “queries and their corresponding results.” See Section V.A.2; see also Sareen, [0027]. Sareen discloses that the stored data “may be a set of recent queries” or “a set of common queries,” both of which a skilled artisan would understand to refer to queries that previously were received at the computer database system. EX1003, ¶135 (citing Sareen, [0027], [0032], [0034]). A skilled artisan would understand Sareen’s disclosure of storing this information in local storage to teach “*storing in memory data about previously received queries at the database system.*” EX1003, ¶135.

3. Retrieving Data from the Memory

Claim 8 recites “*at a server, retrieving selected data from the memory including query data representative of previously submitted queries and information based on frequency counts for the respective previously submitted queries.*” The combination of Sareen and Anick renders this limitation obvious. EX1003, ¶¶136-138.

Sareen discloses local storage that may store both “the most recent queries” submitted by a user and a set of “frequently requested queries” (*i.e.*, “*query data representative of previously submitted queries and information based on frequency counts for the respective previously submitted queries*”). Sareen, [0031]-[0032], [0038]; EX1003, ¶137. Sareen further discloses that its data structure may comprise an augmented trie structure, including “possible portions of a query along with additional information.” Sareen, [0033]. “The information related to each portion of the query may for example be a list of a predetermined number of most frequently requested queries beginning with the portion of the query” (*i.e.*, “*information based on frequency counts for the respective previously submitted queries*”). Sareen, [0033]; EX1003, ¶137. A skilled artisan would understand that this information is stored in and retrieved from the local storage (*i.e.*, “*retrieved . . . from the database*”). EX1003, ¶138.

4. Constructing Indexes

Claim 8 recites “*at the server, constructing indexes using the retrieved selected data*.” The combination of Sareen and Anick renders this limitation obvious. EX1003, ¶139-140.

As discussed above with respect to Claims 4 and 5, Sareen discloses that the data structure, which may be housed in its own server, uses the historical query data

retrieved from the database to build query indexes (*i.e.*, “*constructing indexes using the retrieved selected data*”). See Sections V.D-V.E; EX1003, ¶140.

5. Receiving a Partial Query

Claim 8 recites “*receiving at the server a partial query submitted by a user at a remote user client system, the submitted partial query including an abbreviated or incomplete search query which is not fully representative of an entire search query desired by the user.*” The combination of Sareen and Anick renders this limitation obvious. EX1003, ¶¶141-144.

Sareen discloses “*receiving at the server a partial query submitted by a user at a remote user client system*” for the reasons discussed above with respect to Claim 1. See Section V.A.3; EX1003, ¶142.

Sareen also discloses “*the submitted partial query including an abbreviated or incomplete search query which is not fully representative of an entire search query desired by the user*” for the reasons discussed above with respect to Claim 1. See Section V.A.4; EX1003, ¶143.

To the extent the Board finds that Sareen’s disclosure of a portion of a query entered by a user does not teach “*a partial query submitted by a user,*” this limitation would have been obvious in view of Sareen and Anick for the reasons discussed above with respect to Claim 1. See Section V.A.3; EX1003, ¶144.

6. Determining a Full Query

Claim 8 recites “*at the server, using the indexes to determine a full query corresponding to the submitted partial query, the full query being better representative of the entire search query desired by the user.*” The combination of Sareen and Anick renders this limitation obvious. EX1003, ¶145-147.

Sareen discloses “*determin[ing] a full query corresponding to the submitted partial query, the full query being better representative of the entire search query desired by the user*” for the reasons discussed above with respect to Claim 1. See Sections V.A.3-V.A.4; EX1003, ¶146.

A skilled artisan would understand that the “*determin[ation] [of] a full query*” occurred “*at the server, using the indexes to determine a full query*” for the reasons discussed above with respect to Claims 1 and 5. For example, a skilled artisan would understand that the full query in Sareen is determined by the front-end server (*i.e.*, “*at the server*”) using recent queries and frequently requested queries stored in local memory and retrieved by the data structure (*i.e.*, “*using the indexes*”). See Sections V.A.3.c, V.E; EX1003, ¶147.

I. Claim 9

Claim 9 recites “[*t*]he method of claim 8 further comprising communicating the full query from the server to a search engine for subsequent processing of the

full query.” The combination of Sareen and Anick renders this limitation obvious. EX1003, ¶¶148-150.

Sareen discloses that “[w]hen a list of other queries, which may include the portion of the query, is obtained [by the front-end server] from the data structure, a results structure 340 . . . is consulted.” Sareen, [0034]. Sareen discloses that the front-end server may also consult the back-end database 240 “if results for one or more of the queries that include the portion of the query are not located in the results structure.” Sareen, [0035]. A skilled artisan would understand that the “consultation” disclosed in Sareen involves “*communicating the full quer[ies]*” obtained by the front-end server to the database or results structure. EX1003, ¶149.

A skilled artisan would understand that the results structure and back-end database are both part of a “*search engine.*” EX1003, ¶150. A skilled artisan therefore, would understand that the full queries obtained by the front-end server in Sareen are communicated from the server to the search engine (“*communicating the full query from the server to a search engine for further processing*”). EX1003, ¶150.

J. Claim 10

Claim 10 recites “[t]he method of claim 8 further comprising: *submitting the full query from the server to a search engine; receiving search results at the server from the search engine; and communicating a web page including at least some of*

the search results from the server to the user.” The combination of Sareen and Anick renders this limitation obvious. EX1003, ¶¶151-154.

As discussed with respect to Claim 9, Sareen discloses “*submitting the full query from the server to a search engine.*” See Section V.I; EX1003, ¶152. Sareen also discloses receiving search results at the server from the server. For example, Sareen discloses that its front-end server (*i.e.*, “*server*”) consults the results structure and back-end database (*i.e.*, part of the “*search engine*”) to obtain search results (*i.e.*, “*receiv[es] search results*”). EX1003, ¶153; *see also* Section V.C. And a skilled artisan would understand that Sareen discloses “*communicating a web page including at least some of the search results from the server to the user*” for at least the reasons discussed above with respect to Claim 3. See Section V.C; EX1003, ¶154.

K. Claim 11

Claim 11 recites “[*t*]he method of claim 10 further comprising: at the server, using the indexes to determine one or more alternative queries that match the submitted partial query, and wherein communicating the web page comprises communicating a web page including at least some of the one or more alternative queries.” The combination of Sareen and Anick renders this limitation obvious. EX1003, ¶¶155-160.

Sareen discloses “*using the indexes to determine one or more alternative queries that match the submitted partial query*” for at least the reasons discussed above with respect to Claim 8. *See* Section V.H.6; EX1003, ¶156. For example, Sareen discloses that a portion of a query is received and compared against indexes of previously submitted queries to determine full queries that begin with that portion of a query. *See* Section V.H.6; EX1003, ¶156. Specifically, Sareen explains that “[o]nce the front-end server 230 receives the portion of the query, the front-end server 230 may be responsible for performing a search in order to obtain other queries that have the same portion as the portion of the query.” Sareen, [0031]. A skilled artisan therefore, would understand Sareen to disclose an embodiment in which the computer database system determines multiple full queries that match the portion of the query submitted by the user, one of the queries being the “*full query*” and the remaining queries being “*one or more alternative queries that match the submitted partial query*.” EX1003, ¶156.

Although Sareen discloses presenting search results for at least some of these alternative queries, it does not expressly disclose communicating a web page listing the alternative queries themselves. It would have been obvious, however, to include that functionality within Sareen’s system at least based on the teachings of Anick. EX1003, ¶¶157-160.

Anick discloses returning to the user “a subset of candidate terms (search refinement suggestions)” (*i.e.*, “*one or more alternative queries*”) along with a “listing of search results (top-ranked documents in an initial list of ranked documents).” Anick, 7:53-56, 8:2-7. For example, Anick discloses returning several alternative queries for the partial query “space shuttle,” including “Space Shuttle Challenger” and “Space Shuttle Columbia.” Anick, Fig. 2. These alternative queries are displayed on the user’s interface, such as through a webpage on the user’s search engine browser (*i.e.*, are “*a web page including at least some of the one or more alternative queries*”). *See* Anick, Fig. 2, 8:2-3; EX1003, ¶158.

A skilled artisan would have been motivated to combine this functionality with Sareen’s system to aid the user in efficiently refining his or her query to obtain more useful results. EX1003, ¶159. For example, Anick discloses that its “list of suggested terms [] can be used to improve an initial query.” Anick, 8:34-36. Anick likewise suggests that the suggested queries may “more closely match[] the interests of the user.” Anick, 8:43-47. A skilled artisan would understand that presenting the candidate terms for the alternative queries, rather than just the search results, would allow “a user [to] build an improved query without analyzing documents (or indicia thereof) within the initial group of ranked documents.” EX1003, ¶159 (citing Anick, 8:43-47). In other words, a skilled artisan would understand that incorporating into Sareen’s system the functionality disclosed in Anick would limit the “need to

determine why an initial query produced too many (or too few) results or results that are not directly related to the informational needs of the user.” EX1003, ¶159 (citing Anick, 8:47-51).

A skilled artisan would have had a reasonable expectation of success in making this combination. EX1003, ¶160. Sareen’s system already determines alternative queries. EX1003, ¶160 (citing Sareen, [0031]). Sareen’s system likewise identifies and presents results corresponding to these alternative queries. *See* Sareen, [0034]. Extending this functionality to present to the user the alternative queries themselves, as disclosed by Anick, would have been well within the ordinary abilities of a skilled artisan. EX1003, ¶160.

VI. GROUND 2: CLAIMS 12-14 ARE OBVIOUS IN VIEW OF SAREEN IN COMBINATION WITH ANICK AND ARAVAMUDAN.

A. Claim 12

Claim 12 recites “[t]he method of claim 8 wherein receiving the partial query comprises interpreting the submitted partial query as a series of words separated by delimiters.” The combination of Sareen, Anick and Aravamudan renders this limitation obvious. EX1003, ¶¶161-164.

The combination of Sareen and Anick renders obvious the limitations of claim 8 for at least the reasons discussed above. *See* Section V.H; EX1003, ¶162. To the extent Sareen does not expressly disclose “*interpreting the submitted partial query as a series of words separated by delimiters*,” it would have been obvious to

implement that functionality into Sareen's system in view of at least the teachings of Aravamudan. EX1003, ¶162-164.

Aravamudan discloses a "process of the user starting a new search, entering characters and arriving at the desired result." Aravamudan, [0033]. Aravamudan further discloses a user "enter[ing] a search string character at 401, which could be a variable size prefix of the intended query" (*i.e.*, "*a submitted partial query*"). Aravamudan, [0033]; EX1003, ¶163. For example, Aravamudan discloses entering "B P", "BR P," "B PI," etc. as a partial query representing the user's desired query of "Brad Pitt." Aravamudan, [0033]. Aravamudan discloses that its system then "dynamically retrieves results at 402 for the cumulative substring of characters the user enters and orders the collected results based on (i) the relevance of the matched terms (explained in further detail below), and (ii) the number of characters the user entered." Aravamudan, [0033]. A skilled artisan therefore, would understand Aravamudan to disclose interpreting the submitted query of "B [space] P" as separate words indicating the full query "Brad Pitt" (*i.e.*, "*interpreting the submitted partial query as a series of words separated by delimiters*"). EX1003, ¶163.

A skilled artisan would have been motivated to combine with a reasonable expectation of success this functionality with Sareen's system to more efficiently reconstruct a user's query and to obtain more relevant results. EX1003, ¶164. Sareen already discloses processing a user's partial query to identify "the proper

portion of the query” to be searched. Sareen, [0039] (“For example, if a user enters such queries beginning with “http” and “www,” the user typically does not want to search these queries but may instead be interested in the characters following these frequently-occurring entries.”). Recognizing a user’s submission of a delimiter, such as a space, as an indication that the partial query is a series of words as disclosed in Aravamudan, would be a natural extension of Sareen’s focus on identifying the proper partial query to be searched. EX1003, ¶164. And Aravamudan expressly contemplates that its search functionality is applicable to other environments, such as the computer database system disclosed in Sareen. Aravamudan, [0024].

B. Claim 13

Claim 13 recites “[t]he method of claim 12 further comprising: at the server, interpreting each word of the received partial query as a partial word of the entire search query desired by the user; and using the indexes, determining a full query that matches the partial words specified in the partial query.” The combination of Sareen, Aravamudan and Anick renders this limitation obvious. EX1003, ¶¶165-168.

Aravamudan discloses “interpreting each word of the received partial query as a partial word of the entire search query desired by the user” and “determining a full query that matches the partial words specified in the partial query” for at least the reasons discussed above with respect to Claim 12. See Section VI.A; EX1003,

¶166. For example, Aravamudan discloses interpreting the submitted query of “B [space] P” as separate partial words matching the full query “Brad Pitt.” Aravamudan, [0033].

A skilled artisan would understand that Aravamudan’s disclosed functionality of “*determining a full query that matches the partial words specified in the partial query*” would be implemented in Sareen using Sareen’s indexes (*i.e.*, “*using the indexes*”). EX1003, ¶167. As discussed above with respect to Claim 8, Sareen discloses constructing indexes and using the indexes to determine a full query corresponding to the submitted partial query. *See* Section V.H; EX1003, ¶167. A skilled artisan would understand that full queries matching multi-word partial queries, such as those disclosed in Aravamudan, would be processed in the same way. EX1003, ¶167. A skilled artisan would have been motivated to combine Sareen and Aravamudan for the reasons discussed above with respect to Claim 12. *See* Section VI.A; EX1003, ¶168.

C. Claim 14

Claim 14 recites “[t]he method of claim 13 wherein determining a full query comprises determining the full query that is the most frequently submitted known query, as defined by the indexes, that matches the partial words of the received partial query in the same order.” The combination of Sareen, Aravamudan, and Anick renders this limitation obvious. EX1003, ¶¶169-172.

As discussed above with respect to Claim 2, Sareen discloses “*determin[ing] as the full query the most frequently submitted, previously-submitted query . . . that matches partial words of the partial query.*” See Section V.B.; EX1003, ¶170. As discussed above with respect to Claim 13, a skilled artisan would understand the combination of Sareen, Aravamudan and Anick to determine full queries using the indexes disclosed in Sareen. See Section VI.B; EX1003, ¶171.

Aravamudan discloses returning “Brad Pitt” as the full query for the partial words “B P,” “BR P,” and “B PI” (“*match[ing] the partial words in the same order*”). Aravamudan, [0033]. A skilled artisan would have understood that this functionality, as implemented in Sareen’s system would return a full query “*that matches the partial words of the received partial query in the same order.*” EX1003, ¶172.

VII. GROUND 3: CLAIMS 15-18 ARE OBVIOUS IN VIEW OF SAREEN IN COMBINATION WITH KAMVAR, ARAVAMUDAN AND ANICK.

A. Claim 15

1. Preamble

The preamble of Claim 15 recites “[*a*] computer implemented method for a computer database system.” To the extent the preamble is limiting, the combination of Sareen, Kamvar, Aravamudan and Anick renders this limitation obvious at least for the reasons discussed above with respect to Claim 8. See Section V.H.1; EX1003, ¶173.

2. Storing in Memory

Claim 15 recites “*storing in memory data about previously received queries at the computer database system, the stored data including respective received queries and respective weight values determined based on receipt frequency of the respective received queries.*” The combination of Sareen, Kamvar, Aravamudan and Anick renders this limitation obvious. EX1003, ¶¶174-180.

Sareen discloses “*storing in memory data about previously received queries at the computer database system, the stored data including respective received queries*” for the reasons discussed above with respect to Claim 8. See Section V.H.2; EX1003, ¶175.

Sareen also discloses that the stored data is assigned “*respective weight values determined based on receipt frequency of the respective received queries.*” EX1003, ¶176. For example, Sareen discloses that its data structure may comprise “a list of a predetermined number of most frequently requested queries beginning with the portion of the query.” Sareen, [0033]. Sareen further explains that this “list may be sorted in a descending order of frequency, so that the most common requested query beginning with the portion of the query is first listed” (*i.e.*, according to “*respective weight values determined based on receipt frequency*”). Sareen, [0033].

To the extent the claims require quantitative weights to be stored in memory, it would have been obvious to modify Sareen’s system to implement that

functionality in view of at least the teachings of Kamvar. EX1003, ¶¶177. Kamvar discloses storing an index of previously submitted search queries, where “each of the entries includes a term portion and a popularity value.” Kamvar, [0042]. “The popularity value 605 is a value that indicates how popular the associated term portion may be at any given moment in time. The popularity value 605 can be used to rank the predicted search queries according to current popularity.” Kamvar, [0042]. A skilled artisan would understand the term “popularity value,” as used in Kamvar, to be “*determined based on receipt frequency.*” EX1003, ¶178 (citing Kamvar [0054], [0069]).

A skilled artisan would have been motivated to make this combination to “increase the likelihood that the user will be presented with an intended query term.” Kamvar, [0042]; EX1003, ¶179. Sareen’s system already discloses a system capable of presenting a list of previously submitted queries in order of relative frequency. Sareen, [0033]. A skilled artisan would have recognized that storing in memory a quantitative frequency weight for each term could increase the likelihood that the system presents the user with an intended query. EX1003, ¶179.

A skilled artisan would have had a reasonable expectation of success in making this combination. EX1003, ¶180. As discussed above, Sareen’s system already evaluates the frequency of query submissions and orders the previously submitted queries accordingly. Sareen, [0033]. A skilled artisan would have

understood this process to require that some quantitative measure of frequency be associated with each term. EX1003, ¶180. It would have been well within the ordinary abilities of a skilled artisan to store this quantitative measure of frequency in memory as disclosed by Kamvar. EX1003, ¶180.

3. Receiving a Partial Query

Claim 15 recites “*receiving at a server of the computer database system a partial query submitted by a user at a remote user client system, the submitted partial query including an abbreviated or incomplete search query which is not fully representative of an entire search query desired by the user.*” The combination of Sareen, Kamvar, Aravamudan and Anick renders this limitation obvious for at least the reasons discussed above with respect to Claim 8. *See* Section V.H.5; EX1003, ¶181.

4. Interpreting Each Word of the Partial Query as a Partial Word

Claim 15 recites “*at the server, interpreting each word of the received partial query as a partial word of the entire search query desired by the user.*” The combination of Sareen, Kamvar, Aravamudan and Anick renders this limitation obvious for at least the reasons discussed above with respect to Claim 13. *See* Section VI.B; EX1003, ¶182.

5. Identifying Matching Queries

Claim 15 recites “*at the server, identifying in the memory previously received queries that match the partial words specified in the partial query.*” The combination of Sareen, Kamvar, Aravamudan and Anick renders this limitation obvious. EX1003, ¶¶183-185.

As discussed above with respect to Claim 13, the combination of Sareen, Kamvar, Aravamudan and Anick renders obvious “*identifying . . . previously received queries that match the partial words specified in the partial query.*” See Section VI.B; EX1003, ¶184. As discussed above with respect to Claims 8 and 13, a skilled artisan also would understand that the process of “*identifying. . . previously received queries*” involves consulting query indexes, which are stored “*in memory.*” See Sections V.H.6; VI.B; EX1003, ¶185.

6. Selecting a Full Query

Claim 15 recites “*at the server, selecting as a full query a matching query which is most popular based on the weight values, the full query being better representative of the entire search query desired by the user.*” The combination of Sareen, Kamvar, Aravamudan and Anick renders this limitation obvious. EX1003, ¶¶186-189.

As discussed above with respect to Claim 2, Sareen discloses “*selecting as the full query a matching query which is most popular.*” See Section V.B; EX1003, ¶187.

As discussed above with respect to Claim 7, a skilled artisan would understand the “most frequently requested queries” to be the “*most popular based on the weight values.*” See Section V.G; see also Section VII.A.2; EX1003, ¶188. Indeed, a skilled artisan would understand that the determination of the most popular matching query in the combination of Sareen, Kamvar, Aravamudan and Anick would be made based on the “*respective weight values*” stored in memory according to Kamvar’s disclosure. See Section VII.A.2; EX1003, ¶188.

As discussed above with respect to Claim 8, Sareen discloses “*the full query being better representative of the entire search query desired by the user.*” See Sections V.H.6; EX1003, ¶189.

B. Claim 16

Claim 16 recites “[t]he method of claim 15 further comprising: submitting the full query from the server to a search engine; receiving search results at the server from the search engine; and communicating a web page including at least some of the search results from the server to the user.” The combination of Sareen, Kamvar, Aravamudan and Anick renders this limitation obvious for at least the reasons discussed above with respect to Claim 10. See Section V.J; EX1003, ¶190.

C. Claim 17

Claim 17 recites “[t]he method of claim 16 further comprising: at the server, based on the weights, determining one or more alternative queries that match the submitted partial query, and wherein communicating the web page comprises communicating a web page including at least some of the one or more alternative queries.” The combination of Sareen, Kamvar, Aravamudan and Anick renders this limitation obvious for at least the reasons discussed above with respect to Claim 11. See Section V.K; EX1003, ¶191. As discussed above with respect to Claim 15, a skilled artisan would understand that the determination of multiple full queries, including the claimed “full query” and the “one or more alternative queries” in the combination of Sareen, Kamvar, Aravamudan and Anick would be made based on the “respective weight values” stored in memory according to Kamvar’s disclosure. See Sections VII.A.6.

D. Claim 18

Claim 18 recites “[t]he method of claim 17 wherein communicating the web page comprises communicating a web page with one or more hyperlinks corresponding to the one or more alternative queries, the hyperlinks being selectable by the user viewing the web page to select one of the one or more alternative queries.” The combination of Sareen, Kamvar, Aravamudan and Anick renders this limitation obvious. EX1003, ¶¶192-194.

As discussed above with respect to Claim 11, Anick discloses communicating a web page including at least some of the one or more alternative queries. *See* Section V.K; EX1003, ¶193. Anick further discloses displaying the alternative queries along with two selectable “tags” (*i.e.*, “a hyperlink corresponding to the one or more alternative queries.” Anick, 8:3-5; *see also id.* at Fig. 2. For example, Anick discloses that “[w]hen the user presses a tag 138, the term 136 that corresponds to the tag 138 is added to initial query 132 and the whole process repeats with the new query.” Anick, 8:10-12. Anick likewise explains that “[w]hen the user presses another tag 139, the term 135 that corresponds to the tag 138 replaces the initial query 132 and the search engine server process that term 136 as the new query.” Anick, 8:13-16.

A skilled artisan would have been motivated to combine this functionality with Sareen’s system for the reasons discussed above with respect to claim 11. *See* Section V.K; EX1003, ¶194. Indeed, a skilled artisan would have been motivated to include in the displayed webpage the selectable tags of Anick to further aid the user in quickly refining his or her search without having to enter additional text. EX1003, ¶194.

VIII. The Board Should Institute *Inter Partes* Review

The Board shall not deny institution based on the *Fintiv* factors “where a petition presents compelling evidence of unpatentability.” Memorandum from

Director Vidal, Interim Procedure for Discretionary Denials in AIA Post-Grant Proceedings with Parallel District Court Litigation, 2 (USPTO June 21, 2022). Here, the Board need not apply the *Fintiv* factors because the evidence presented in Grounds 1-3, would plainly lead to a conclusion that one or more claims are unpatentable by a preponderance of the evidence.

Moreover, all *Fintiv* factors favor institution or are neutral. Factor 1 is neutral because no party has requested a stay. *See, e.g., Sand Revolution II, LLC v. Cont'l Intermodal Grp.-Trucking LLC*, IPR2019-01393, Paper 24, at 7 (June 16, 2020) (informative). Factor 2 favors institution because no trial date is scheduled in either R2's suit against Petitioner American Airlines, Inc. ("American Airlines Litigation") (EX1016, 4) or the Hilton Litigation³, and any comparison to a projected final written decision would require speculation. The Board routinely institutes in such circumstances, even if trial is scheduled before issuance of a final written decision. *See, e.g., Sand Revolution*, IPR2019-01393, Paper 24, at 7-14 (five months before); *Netnut Ltd., v. Bright Data Ltd.*, IPR2021-01492, Paper 12, at 9-16 (Mar. 21, 2022) (six months before, no stipulation); *Volkswagen Group of Am., Inc., v. Arigna Tech. Ltd.*, IPR2021-01321, Paper 10, at 7-9 (Feb. 15, 2022) (three months before, no stipulation).

³ A scheduling order has not yet been entered in the Hilton Litigation.

Factor 3 favors institution because the parties have expended almost no resources in the Hilton litigation and only a small percentage of the resources to be expended in the American Airlines Litigation. Petitioners acted expeditiously by filing the Petition in the early stages of fact discovery and before expert discovery in the American Airlines Litigation and prior to a scheduling order and within four months of the answer being filed in the Hilton Litigation. *See* EX1016, 2-3; *Sotera Wireless, Inc. v. Masimo Corp.*, IPR2020-01019, Paper 12, at 16-17 (Dec. 1, 2020) (precedential). Factor 4 favors institution because Petitioners challenge all claims of the '317 patent, while Patent Owner asserts only a subset in the litigation. EX1017, 27; EX1018, 2. Petitioners also stipulate that, if the Petition is instituted, Petitioners will not pursue the grounds identified in the Petition in litigation. *Sand Revolution*, IPR2019-01393, Paper 24, at 11-12. Factor 5 favors institution because, while Petitioners are the defendants in the parties' respective litigations, Patent Owner has asserted the '317 patent in over a dozen different litigations since January 2021. EX1019. Institution thus, will promote judicial efficiency and integrity by reducing the number issues to be addressed in numerous parallel litigations. Factor 6 favors institution because Petitioners have set forth a strong case of unpatentability. *Sand Revolution*, IPR2019-01393, Paper 24, at 13.

Finally, Petitioners' challenges do not advance "the same or substantially the same prior art or arguments previously ... presented to the Office." *See* 35 U.S.C.

§ 325(d); *Becton, Dickinson & Co. v. B. Braun Melsungen AG*, IPR2017-01586, Paper 8 (Dec. 15, 2017) (precedential: §III.C.5, ¶1). The Examiner had neither Sareen nor Anick, nor any similar reference, and also did not have the opinions of Dr. Bederson.

IX. Mandatory Notices Under 37 C.F.R. § 42.8

A. Real Party-in-Interest

The real parties-in-interest for this Petition are American Airlines, Inc. and Hilton Domestic Operating Company, Inc.

B. Related Matters

To the best of Petitioners' knowledge, the '317 patent is or has been involved in the following proceedings:

Name	No.	Court	Filed	Status
<i>R2 Solutions LLC v. 7-Eleven, Inc.</i>	3:22-cv-2868	N.D. Tex.	Dec. 21, 2022	Pending
<i>R2 Solutions LLC v. Southwest Airlines Co.</i>	3:22-cv-2869	N.D. Tex.	Dec. 21, 2022	Pending
<i>R2 Solutions LLC v. GameStop Corp. et al</i>	3:22-cv-2870	N.D. Tex.	Dec. 21, 2022	Closed
<i>R2 Solutions LLC v. Hilton Domestic Operating Company Inc.</i>	3:22-cv-2761	N.D. Tex.	Dec. 12, 2022	Pending

<i>R2 Solutions LLC v. American Airlines, Inc.</i>	4-22-cv-00353	E.D. Tex.	Apr. 28, 2022	Pending
<i>R2 Solutions LLC v. CVS Health Corporation et al.</i>	4-22-cv-00354	E.D. Tex.	Apr. 28, 2022	Closed
<i>R2 Solutions LLC v. Hilton Worldwide Holdings Inc. et al.</i>	4-22-cv-00356	E.D. Tex.	Apr. 28, 2022	Closed
<i>R2 Solutions LLC v. Citigroup Technology, Inc.</i>	4-22-cv-00357	E.D. Tex.	Apr. 28, 2022	Closed
<i>R2 Solutions LLC v. Booking.com BV</i>	4-21-cv-00942	E.D. Tex.	Nov. 29, 2021	Closed
<i>R2 Solutions LLC v. Booking.com Transport Limited</i>	4-21-cv-00943	E.D. Tex.	Nov. 29, 2021	Closed
<i>R2 Solutions LLC v. Agoda Company Pte. Ltd.</i>	4-21-cv-00944	E.D. Tex.	Nov. 29, 2021	Closed
<i>R2 Solutions LLC v. Redfin Corporation</i>	4-21-cv-00945	E.D. Tex.	Nov. 29, 2021	Closed
<i>R2 Solutions LLC v. Roku, Inc.</i>	6-21-cv-00553	W.D. Tex.	June 1, 2021	Closed
<i>R2 Solutions LLC v. Samsung Electronics America, Inc.</i>	4-21-cv-00089	E.D. Tex.	Jan. 29, 2021	Closed
<i>R2 Solutions LLC v. Deezer SA</i>	4-21-cv-00090	E.D. Tex.	Jan. 29, 2021	Closed

C. Lead and Backup Counsel

Lead Counsel	Backup Counsel
John B. Campbell Reg. No. 54,665 jcampbell@McKoolSmith.com McKool Smith, P.C. 303 Colorado, Suite 2100 Austin, Texas 78701 Telephone: (512) 692-8700 Facsimile: (512) 692-8744	Geoffrey Smith (to be admitted <i>pro hac vice</i>) gsmith@McKoolSmith.com McKool Smith, P.C. 303 Colorado, Suite 2100 Austin, Texas 78701 Telephone: (512) 692-8700 Facsimile: (512) 692-8744 Raphael Chabaneix Reg. No. 78,922 rhabaneix@McKoolSmith.com McKool Smith, P.C. 300 Crescent Court, Suite 1500 Dallas, Texas 75201 Telephone: (214) 978-4000 Facsimile: (214) 978-4044 Steven Peters Reg. No. 73,193 speters@McKoolSmith.com McKool Smith, P.C. 1999 K Street, NW Suite 600 Washington, DC 20006 Telephone: (202) 370-8300 Facsimile: (202) 370-8344 David E. Finkelson (to be admitted <i>pro hac vice</i>) dfinkelson@mcguirewoods.com

	McGuireWoods LLP 800 East Canal Street Richmond, VA 23219-3916 Telephone: (804) 775-1000 Facsimile: (804) 698-2016 Kyle Smith Reg. No. 78,579 ksmith@mcguirewoods.com McGuireWoods LLP 501 Fayetteville St. Raleigh, NC 27601 Telephone: (919) 755-6600 Facsimile: (919) 755-6699
--	---

D. Service Information

Please address all correspondence to lead and backup counsel at the addresses shown above. Petitioners also consent to electronic service by e-mail.

X. Grounds for Standing

Petitioners certify that the '317 patent is available for *inter partes* review and that Petitioners are not barred or estopped from requesting *inter partes* review to challenge the claims on the grounds identified herein. Petitioners filed this Petition within one year of service of the original complaints against Petitioners.

XI. Conclusion

Petitioners have established a reasonable likelihood of prevailing with respect to the challenged claims and request institution of *inter partes* review and cancellation of said claims.

Date: April 27, 2023

Respectfully submitted,

/s/ John B. Campbell

John Campbell, *Lead Counsel*

Reg. No. 54,665

Telephone: (512) 692-8700

Facsimile: (512) 692-8744

jcampbell@McKoolSmith.com

McKOOL SMITH, P.C.

303 Colorado, Suite 2100

Austin, Texas 78701

Counsel for Petitioners

CERTIFICATE OF COMPLIANCE

Pursuant to 37 C.F.R. § 42.24(a)(i), the undersigned certifies that this Paper—
exclusive of the table of contents, table of authorities, grounds for standing under §
42.104, mandatory notices under § 42.8, certificate of service, and this certificate of
word count—includes 12,516 words.

Date: April 27, 2023

/John B. Campbell/
John B. Campbell, Lead Counsel
Reg. No. 54,665

CERTIFICATE OF SERVICE

The undersigned certifies that, in accordance with 37 C.F.R. § 42.6(e) and 37 C.F.R. § 42.105, **Petition for *Inter Partes* Review of U.S. Patent No. 8,209,317, the associated Power of Attorney, and associated exhibits were served via FedEx Priority Overnight® on the correspondence address of record with the USPTO as follows:**

Crowell
PO Box 10395
Chicago, IL 60610

A courtesy copy is also being sent via FedEx Priority Overnight® to litigation counsel at:

Edward R. Nelson, III
Brent N. Bumgardner
Carder W. Brooks
Christopher G. Granaghan
John P. Murphy
Nelson Bumgardner Conroy PC
3131 W. 7th Street, Suite 300
Fort Worth, TX 76107
Telephone: (817) 377-9111
ed@nelbum.com
brent@nelbum.com
carder@nelbum.com
chris@nelbum.com
murphy@nelbum.com

Dated: April 27, 2023

By: /John B. Campbell/
John B. Campbell, Lead
Counsel
Reg. No. 54,665