

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AMAZON WEB SERVICES, INC.,
Petitioner,

v.

KOVE IO, INC.,
Patent Owner.

Case IPR2020-00276
Patent 7,233,978

PETITION FOR
INTER PARTES REVIEW OF U.S. PATENT NO. 7,233,978

Filing Date: December 12, 2019

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	BACKGROUND OF THE '978 PATENT-AT-ISSUE	3
A.	The Domain Name System (DNS).....	3
B.	The '978 Patent	4
C.	Relationship Between '978 Patent and DNS	8
D.	Prosecution History	10
E.	Priority Date	11
F.	Level of Ordinary Skill	11
III.	IDENTIFICATION OF PRIOR ART	12
A.	Network Working Group RFC1034 (Ex. 1003).....	12
B.	U.S. Patent No. 5,542,087 to Neimat (Ex. 1004).....	13
C.	“Load Management in Distributed Video Servers” by N. Venkatasubramanian (Ex. 1005).....	13
D.	Other Evidence of the State of the Art (Ex. 1006-1028)	14
IV.	IDENTIFICATION OF CHALLENGE AND RELIEF REQUESTED [37 C.F.R. §§42.104(B)(1)-(2)]	14
V.	CLAIM CONSTRUCTION [37 C.F.R. § 42.104(B)(3)]	15
VI.	GROUND OF UNPATENTABILITY [37 C.F.R. §§ 42.104(B)(4)-(5)] ..	17
A.	Ground 1: RFC1034 Anticipates Claims 1, 3, 6, 10, 14, and 31.	17
1.	Overview of Mapping Between '978 Patent and RFC1034	17
2.	Independent Claim 1	21
3.	Dependent Claims 3 and 6	30

4.	Independent Claim 10	32
5.	Independent Claim 14	36
6.	Independent Claim 31	38
B.	Ground 2: RFC1034 Renders Obvious Claim 6.	39
C.	Ground 3: RFC1034 in view of Neimat Renders Obvious Claims 17, 23, and 30.....	42
1.	RFC1034 and Neimat, in combination, disclose all limitations in Independent Claim 17 and Dependent Claims 23 and 30.....	42
2.	One of Ordinary Skill Would Have Had Reason to Combine RFC1034 with Neimat.	53
D.	Ground 4: The Combination of RFC1034, Neimat, and Venkatasubramanian Renders Obvious Claim 24.	57
VII.	LACK OF SECONDARY CONSIDERATIONS EVIDENCE.....	61
VIII.	MANDATORY NOTICES	61
A.	Real Party-in-Interest [37 C.F.R. §42.8(b)(1)].....	61
B.	Related Matters [37 C.F.R. §42.8(b)(2)].....	62
C.	Lead/Back-up Counsel [37 C.F.R. §42.8(b)(3)]	62
D.	Service Information [37 C.F.R. §42.8(b)(4)]	62
IX.	GROUND FOR STANDING [37 C.F.R. §42.104(A)].....	62
X.	CONCLUSION.....	62

TABLE OF AUTHORITIES

	Page(s)
CASES	
<i>Activision Blizzard, Inc. v. Acceleration Bay, LLC</i> , Case IPR2015-01972 (P.T.A.B. Mar. 23, 2017)	13
<i>DyStar Textilfarben GmbH v. C.H. Patrick Co.</i> , 464 F.3d 1356 (Fed. Cir. 2006)	43
<i>Ericsson, Inc. v. Intellectual Ventures I LLC</i> , Case IPR2014-00527 (P.T.A.B. May 18, 2015).....	14
<i>Game & Tech. Co. v. Activision Blizzard Inc.</i> , 926 F.3d 1370 (Fed. Cir. 2019)	40, 41
<i>In re Translogic Tech., Inc.</i> , 504 F.3d 1249 (Fed. Cir. 2007)	39
<i>Kimberly-Clark Worldwide, Inc. v. First Quality Baby Prods., LLC</i> , 431 F. App'x 884 (Fed. Cir. 2011)	40
<i>KSR Int'l Co. v. Teleflex Inc.</i> , 550 U.S. 398 (2007).....	39, 56
<i>Mangrove Partners Master Fund, Ltd. v. Virnetx Inc.</i> , Case IPR2015-01046 (P.T.A.B. Sept. 9, 2016).....	12
<i>Nidec Motor Corp. v. Zhongshan Broad Ocean Techs. LLC</i> , 868 F.3d 1013 (Fed. Cir. 2017)	15
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1302 (Fed. Cir. 2015) (en banc)	16
<i>Research Corp. Techs., Inc. v. Microsoft Corp.</i> , 627 F.3d 859 (Fed. Cir. 2010)	11
<i>SZ DJI Tech. Co., v. Synergy Drone, LLC</i> , Case IPR2018-00205 (P.T.A.B. Mar. 22, 2018)	40
<i>Wyers v. Master Lock Co.</i> , 616 F.3d 1231 (Fed. Cir. 2010)	61

<i>ZUP, LLC v. Nash Mfg., Inc.</i> , 896 F.3d 1365 (Fed. Cir. 2018)	40
--	----

STATUTES

35 U.S.C. § 102(a)	12, 13
35 U.S.C. § 102(b)	13, 14
35 U.S.C. § 103(a)	14, 15
35 U.S.C. § 311(b)	13

OTHER AUTHORITIES

M.P.E.P. § 2143(I)(A).....	59
----------------------------	----

PETITIONER EXHIBIT LIST

Exhibit Number	Description
1001	Patent-at-Issue: U.S. Patent No. 7,233,978 to Overton.
1002	Expert Declaration of Jon Weissman Ph.D, Professor of Computer Science, University of Minnesota.
1003	Network Working Group RFC 1034 (“RFC1034”).
1004	U.S. Patent No. 5,542,087 to Neimat (“Neimat”).
1005	“Load Management in Distributed Video Servers” by N. Venkatasubramanian and S. Ramanathan (“Venkatasubramanian”).
1006	“Data Allocation in Distributed Database Systems” by Peter M.G. Apers (“Apers”).
1007	“Distributed Cooperative Web Servers” by Scott M. Baker and Bongki Moon (“Baker”).
1008	“Grapevine: An Exercise in Distributed Computing” by Andrew D. Birrell, Roy Levin, Roger M. Needham, and Michael D. Schroeder (“Birrell”).
1009	“Location-Independent Naming for Virtual Distributed Software Repositories” by Shirley Browne, Jack Dongarra, Stan Green, Keith Moore, Theresa Pepin, Tom Rowan, and Reed Wade (“Browne”).
1010	“Decentralizing a Global Naming Service for Improved Performance and Fault Tolerance” by David R. Cheriton and Timothy P. Mann (“Cheriton”).
1011	“Cooperative Caching: Using Remote Client Memory to Improve File System Performance” by Michael D. Dahlin, Randolph Y. Wang, Thomas E. Anderson, and David A. Patterson (“Dahlin”).
1012	“Design and Implementation of DDH: A Distributed Dynamic Hashing Algorithm” by Robert Devine (“Devine”).

Exhibit Number	Description
1013	“An Adaptive Data Placement Scheme for Parallel Database Computer Systems” by Kien A. Hua and Chiang Lee (“Hua”).
1014	“Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web” by David Karger, Eric Lehman, Tom Leighton, Matthew Levine, Daniel Lewin, and Rina Panigrahy (“Karger”).
1015	“The Core Legion Object Model” by Mike Lewis and Andrew Grimshaw (“Lewis”).
1016	“Distributed Cooperative Apache Web Server” by Quanzhong Li and Bongki Moon (“Li”).
1017	“Linear Hashing : A New Tool for File and Table Addressing” by Witold Litwin (“Litwin1980”).
1018	“LH* - Linear Hashing for Distributed Files” by Witold Litwin, Marie-Anne Neimat, and Donovan A. Schneider (“Litwin1993”).
1019	“Scale in Distributed Systems” by B. Clifford Neuman (“Neuman”).
1020	“The Clearinghouse: A Decentralized Agent for Locating Named Objects in a Distributed Environment” by Derek C. Oppen and Yogen K. Dalal (“Oppen”).
1021	“Hash Routing for Collections of Shared Web Caches” by Keith W. Ross (“Ross”).
1022	“Distributed Web Caching System with Consistent Hashing” by Alexander Sherman (“Sherman”).
1023	“Mariposa: a Wide-Area Distributed Database System” by Michael Stonebraker, Paul M. Aoki, Witold Litwin, Avi Pfeffer, Adam Sah, Jeff Sidell, Carl Staelin, and Andrew Yu (“Stonebaker”).
1024	“An Analysis of Diffusive Load-Balancing” by Raghu Subramanian and Isaac D. Scherson (“Subramanian”).

Exhibit Number	Description
1025	“A Model for Worldwide Tracking of Distributed Objects” by Maarten van Steen, Franz J. Hauck, and Andrew S. Tanenbaum (“van Steen”).
1026	U.S. Patent No. 5,230,047 to Frey (“Frey”).
1027	U.S. Patent No. 6,430,618 to Karger (“Karger618”).
1028	U.S. Patent No. 6,466,980 to Lumelsky (“Lumelsky”).
1029	Network Working Group RFC 1035 (“RFC1035”).
1030	Network Working Group RFC 2026 (“RFC2026”).
1031	Network Working Group RFC 2168 (“RFC2168”).
1032	File History for U.S. Patent No. 7,233,978.
1033	U.S. Patent Application No. 09/111,896.
1034	U.S. Provisional Application No. 60/209,070.
1035	U.S. Provisional Application No. 60/277,408.
1036	IEEE Declaration for Venkatasubmanian Prior Art.
1037	IEEE Declaration for Lee Prior Art.
1038	<i>Kove IO, Inc. v. Amazon Web Services, Inc.</i> , No. 1:18-cv-08175, Complaint (N.D. Ill. Dec. 12, 2018) (Dkt. 1).
1039	<i>Kove IO, Inc. v. Amazon Web Services, Inc.</i> , No. 1:18-cv-08175, Memorandum in Support of Amazon’s Motion to Dismiss (N.D. Ill. April 5, 2019) (Dkt. 38).
1040	<i>Kove IO, Inc. v. Amazon Web Services, Inc.</i> , No. 1:18-cv-08175, Kove’s Memorandum in Opposition to Amazon’s Motion to Dismiss (N.D. Ill. May 13, 2019) (Dkt. 44).

Exhibit Number	Description
1041	<i>Kove IO, Inc. v. Amazon Web Services, Inc.</i> , No. 1:18-cv-08175, Section 101 Hearing Tr. (N.D. Ill. May, 30, 2019).

I. INTRODUCTION

Petitioner Amazon Web Services, Inc. (“AWS”) requests *inter partes* review of claims 1, 3, 6, 10, 14, 17, 23, 24, 30, and 31 of U.S. Patent No. 7,233,978 (“the ’978 patent”), filed June 1, 2001, and assigned to Kove IO, Inc. (“Patent Owner”).

The ’978 patent discloses a network of “location servers,” which store “location information” for data on a distributed computer network. Information regarding the *location* of data (which Patent Owner calls the “where”) is stored separately from the underlying data itself (which Patent Owner calls the “what”). Patent Owner states “[w]hen you separate what we characterize as the where from the what, there is some magic in that.” (See Ex. 1041, 26.)

Patent Owner did not invent a location server network. To the contrary, this concept was well-known long before the ’978 patent: it is the foundation of the Domain Name System (DNS) at the heart of the modern Internet. As shown below, all of the challenged claims are either anticipated by DNS or obvious variations of DNS, as described in the Internet Engineering Task Force Network Working Group Request for Comments 1034 (RFC1034) publication.

Ground 1 shows that RFC1034 anticipates independent claims 1, 10, 14, and 31, and dependent claims 3 and 6. RFC1034 discloses “name servers” that store information about network address locations (the “where,” for example, a

mailbox address) that is separate from the underlying data (the “what” in the mailbox).

Ground 2 shows that, in the alternative, RFC1034 renders obvious claim 6. Any distinction between RFC1034 and claim 6 is so trivial that the claim would have been obvious in view of RFC1034 and the knowledge of one of ordinary skill.

Ground 3 shows that independent claim 17 and dependent claims 23 and 30 would have been obvious in view of RFC1034 and U.S. Patent No. 5,542,087 to Neimat. Claims 17, 23, and 30 describe transferring location information from one server to another when a performance criterion is met. Neimat discloses a “splitting” memory mechanism to address this same kind of performance concern and, in combination with RFC1034, renders obvious claims 17, 23, and 30.

Ground 4 shows that dependent claim 24 would have been obvious. Claim 17 recites a performance criterion. Dependent claim 24 specifies a particular type of performance criterion: a “transaction rate limit.” “Load Management in Distributed Video Servers” by Venkatasubramanian discloses a transaction rate limit. Venkatasubramanian, in combination with RFC1034 and Neimat, renders obvious this subject matter.

There is a high likelihood that Petitioner will prevail with respect to each of the challenged claims. Therefore, the Board should institute trial. Petitioner has

paid the required fee under 37 C.F.R. §§42.15(a) and 42.103 through P.R.P.S. Please assess any fee deficiency or credit to Deposit Account No. 10-0460.

II. BACKGROUND OF THE '978 PATENT-AT-ISSUE¹

A. The Domain Name System (DNS)

The ever-growing demand for powerful data processing systems has fueled the evolution of computers and networks. (*See* Ex. 1002, ¶22; *see also id.* Section V.) Before the 1980s, large databases were often stored on a single computer. With the expansion of the Internet and other developments in network technology, databases have increasingly become distributed, *i.e.*, associated with multiple computers. Accurate processing of database queries in distributed databases is critical.

The Domain Name System (DNS) is a well-known example of a distributed database system. (*Id.*, ¶33.) Indeed, nearly all of the Internet's network services have used DNS since the 1980s. Before DNS, one could only make connections between computers in a network by providing machine readable IP (Internet Protocol) addresses—numerical labels assigned to each device—such as 121.7.106.83. But as the number of host computers on the Internet grew exponentially in the

¹ This background, and the Grounds of Unpatentability herein, are further supported by the Declaration of Petitioner's expert, Jon Weissman, Ph.D., P.E., who has over twenty-five years of experience with the kinds of distributed computer systems at issue here. (Ex. 1002, ¶¶1-9.)

1980s, manually providing IP addresses became unwieldy. Mnemonic domain names, like mail.amazon.com, became preferred to identify hosts. Hence, directory services that mapped domain names to IP addresses became essential.

DNS accomplishes just that. DNS employs a hierarchy of “name servers” that map domain names to IP addresses. (*See id.*, Section X.A.) It also discloses a protocol that allows clients to seek, and receive, the IP address associated with a particular domain name, so that the client can then use that IP address to contact the host. DNS allows a client to use a unique identifier (*e.g.*, a domain name) for an entity (*e.g.*, a host) to retrieve the location information (*e.g.*, IP address) for that entity, and then use that location information to retrieve data from the entity itself.

DNS has been the industry standard for mapping Internet location information to domain names since the 1980s. It is disclosed in a series of Requests for Comments (RFCs) published by the Internet Engineering Task Force Network Working Group, including RFC1034.

B. The '978 Patent

Like DNS, the '978 patent reflects a system that “efficiently tracks the location of data associated with an individual entity in a distributed database.” (Ex. 1001, 7:55-57.) It discloses “location servers” that are “configured to answer the question: Where is data associated with a particular entity in a distributed database

system?” (*Id.*, 5:37-40.) A location server “is only concerned with the location of that data, rather than the endpoint referent data itself.” (*Id.*, 5:45-47.)

The '978 patent coins the acronym “NDTP” to describe its system. NDTP stands for “network distributed tracking protocol” and “is a transfer protocol having the capability to manipulate location of information associated with an individual entity in the distributed database system.” (*Id.*, 4:20-25.) The specification elaborates as follows:

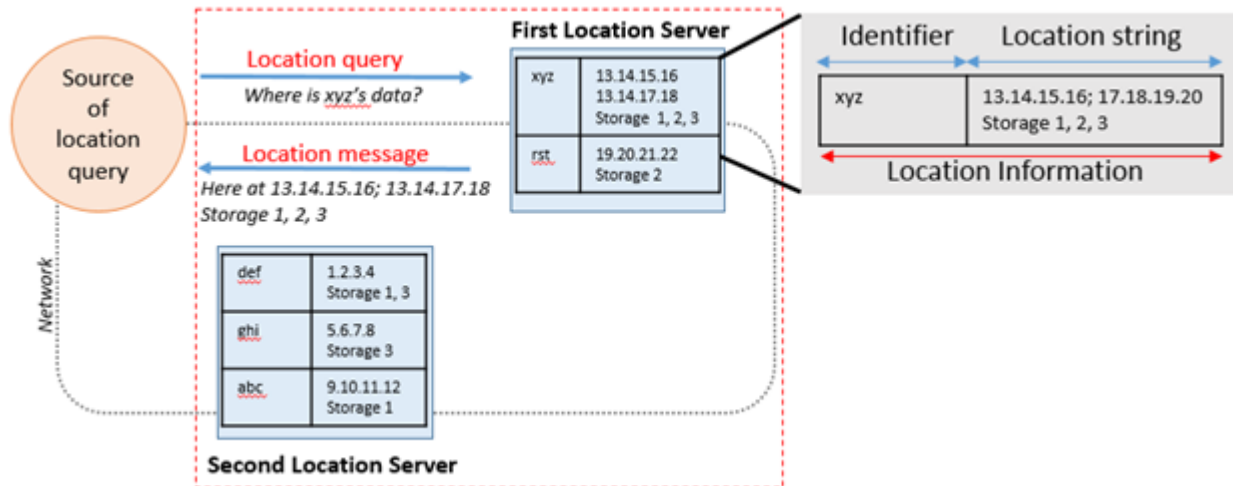
- “[A] ‘server’ is a network-attached component that maintains a set of identifier/location mappings that are modified or returned in response to NDTP request messages from clients.” (*Id.*, 4:34-38.)
- “A ‘data location’ or ‘location’ is an encoding, for example a string, that is a member of a set of associations with an identifier in an NDTP server.” (*Id.*, 4:28-30.)
- “[A]n ‘identifier’ is a unique encoding, which may be a string in one embodiment, with which zero or more data location specifiers are associated in an NDTP server.” (*Id.*, 4:25-28.)
- “[A] ‘client’ is a network-attached component that initiates add, delete, lookup and update of identifier/location mappings, or associations, from an NDTP server with NDTP request messages.” (*Id.*, 4:30-34.)

Representative independent claim 1 describes location servers that provide location information in response to location queries. The location information on one location server differs from the location information on a second location server. The location information has (i) an identifier and (ii) a location string. The system can ask one location server if it has location information associated with a

certain identifier. If that location server has that information, then it returns the location of that information.

U.S. Patent No. 7,233,978, Claim 1
[1.Preamble] A system having a plurality of location servers for managing location information and providing location information to location queries, the system comprising:
[1.a.1] a first location server containing a first set of location information corresponding to at least one entity,
[1.a.2] the location information comprising an identifier and at least one location string associated with the identifier,
[1.a.3] wherein the identifier identifies an entity and the location string specifies a location of data pertaining to the entity
[1.b] a second location server comprising a second set of location information, wherein at least a portion of the second set of location information differs from the first set of location information; and
[1.c.1] programming logic stored on each of the location servers responsive to a location query identifying a desired entity to return a location message,
[1.c.2] the location message comprising at least one location of data pertaining to the desired entity, if the location server receiving the location query contains location information for the desired entity.

Below is a graphical representation (prepared by AWS) of the system described by claim 1:



(Ex. 1002, ¶47.)

In this figure, the identifier “xyz” identifies an entity (such as a mail server) containing data (such as stored e-mail). The location query is sent to a first location server: “where is xyz’s data?” The first location server contains location information, consisting of the identifier (xyz) and the location string or strings corresponding to the entity identified by xyz (in this example, 13.14.15.16; 17.18.19.20). The location server returns a location message, reflecting that the entity identified by xyz is available at 13.14.15.16 and 17.18.19.20. A second location server contains location information associated with other identifiers.

The '978 patent has independent claims 1, 10, 14, 17, and 31. Claim 10 is similar to claim 1, and adds a “transfer protocol,” as well as the limitation that the servers are arranged in a “cluster topology,” “such that each location server contains a unique set of location information of an aggregate set of the location information.” Claim 14 is a method claim similar to claim 1. Claim 17 is a method

claim which describes a system where the location servers rebalance the storage of location information amongst themselves in order to maintain predetermined performance criteria. Claim 31 is similar to claim 1 and adds a “transfer protocol.”

C. Relationship Between ’978 Patent and DNS

The ’978 patent discloses the same kind of system as DNS for retrieving location information. In DNS, the client requests location information associated with an identifier from a location server; the location server returns the location information; and the client uses that location information to contact the entity associated with that identifier.

The ’978 patent concedes as much. It directly equates DNS’s mapping between domain names and addresses to the ’978 patent’s mapping between identifiers and locations:

One way to understand this is as a highly dynamic domain name service (DNS) for data. DNS maintains a mapping between names and machines. NDTP and its associated servers maintain a mapping between identifiers and locations. The identifier/location mapping maintained by NDTP servers is much more dynamic (suitable for frequent updates), than the domain name/IP address mapping maintained by DNS. NDTP may be viewed in this respect as fully generalized name service suitable for any kind of service and not restricted to a specialized service.

(Ex. 1001, 24:2-11.) The specification purports to distinguish the invention from DNS on the ground that it discloses a mapping that is “suitable for frequent up-

dates” and is “not restricted to a specialized service.” (*Id.*) The claims do not recite those purported distinctions.

The provisional applications that led to the '978 patent also note the similarity of NDTP to DNS, while mentioning minor distinctions from DNS that appear nowhere in the claims. The '070 Provisional Application states: “one might view the NDTP server as generalizing the Internet Domain Name Service (DNS). However, the NDTP server provides two critical additional features which DNS does not provide.” (Ex. 1034, 10.) The first is “arbitrary identifiers and locations.” (*Id.*) According to the provisional application, “DNS primarily supports mapping IP domain names to IP addresses, with secondary support for a few other specialized mappings.” (*Id.*) But “more services are being offered through HTTP,” which “often rel[y] on additional location information to connect to a given service, such as a URL.” (*Id.*) IP addresses are “inadequate to locate these rapidly proliferating services,” whereas NDTP offers “location generalization,” “even when data components are migrated from one format to another.” (*Id.*)

The second purported “critical additional feature” is “machine driven updates.” The application states DNS “system administrators usually maintain DNS mappings by hand” and “DNS caching protocols view flushing intervals of around one half hour to be reasonable.” (*Id.*; *see also* Ex. 1035, 8.) Again, this machine driven updates feature is not recited in the claims.

D. Prosecution History

The '978 patent gained allowance based on Patent Owner's broad, and inaccurate, representation that the prior art did not disclose the management of location information separately from data.

On August 22, 2005, the Examiner initially rejected claims as unpatentable over U.S. Patent No. 6,466,980 by Lumelsky (Ex. 1028). In response, Patent Owner argued that "Lumelsky discloses using location information in service of optimizing **distribution of multimedia objects**, whereas Applicants' specification **manages location information** separately from the data to which the location information pertains." (Ex. 1032, 82 (emphasis original).)² The Examiner subsequently issued a notice of allowance stating that the prior art fails to teach "a system in which a location query requests from multiple servers location information, separate from the data objects to which the location information pertains, to be returned to the requestor." (*Id.*, 15.)

² Patent Owner also referred to "the prior art not using the management of location identifier in order to support load balancing." (*Id.*, 45; *see also id.*, 32.)

E. Priority Date

The claims at issue are not entitled to priority to an application before the June 1, 2001 filing date of the '978 patent.³ Although Patent Owner has alleged priority to the July 8, 1998 filing date of U.S. App. No. 09/111,896 (Ex. 1033), Patent Owner has not shown the '896 application contains supporting disclosure of several claim limitations. Claim limitations not supported include, for example, “location string,” “at least a portion of the second set of location information differs from the first set,” “transferring a portion of the identifiers,” and “predetermined performance limit.”

Because at least these limitations were not sufficiently described in the priority chain prior to June 1, 2001, Patent Owner cannot claim priority to an application before that date. *See Research Corp. Techs., Inc. v. Microsoft Corp.*, 627 F.3d 859, 870 (Fed. Cir. 2010) (“Entitlement to a filing date extends only to subject matter that is disclosed.”).

F. Level of Ordinary Skill

Petitioner applies the perspective of one of ordinary skill around the time of the alleged invention of the '978 patent in its analysis of the prior art. Such a person would typically have had a bachelor's degree in computer science (or a similar

³ Regardless, the references used in Grounds 1-4 are all prior art to the purported July 8, 1998 priority date.

field) and at least two years of experience working with distributed software systems, or at least seven years of experience working with distributed systems (or in a similar field), or equivalent. (See Ex. 1002, ¶20.) One of skill would have had knowledge of prior art references, such as RFC1034, as well as the DNS system. (*Id.*)

III. IDENTIFICATION OF PRIOR ART

A. Network Working Group RFC1034 (Ex. 1003)

RFC1034 is the base reference for Grounds 1-4. As detailed in the supporting declaration of Dr. Weissman, RFC1034 was published in November 1987 and was well-known in the art long before the alleged 1998 priority date. (*Id.*, ¶¶60-63.) Dr. Weissman personally studied RFC1034 (and DNS in general) while working on the Legion naming system around 1994-1997. (*Id.*, ¶63.) RFC1034 is prior art under pre-AIA §§ 102(a) and (b) because it was a printed publication as of November 1987, at least one year before any claimed priority date. Indeed, the Board has held that RFC1034 is a printed publication. *Mangrove Partners Master Fund, Ltd. v. Virnetx Inc.*, Case IPR2015-01046, at *24 (P.T.A.B. Sept. 9, 2016) (Paper 71) (“[W]e agree with Petitioner that RFC 1034 is a prior art printed publication.”).

RFC1034 was not before the Examiner during prosecution.

B. U.S. Patent No. 5,542,087 to Neimat (Ex. 1004)

Neimat is a secondary reference for Grounds 3-4 that discloses methods of scaling and splitting. Neimat is prior art under pre-AIA §§ 102(a), (b), and (e) because it issued on July 30, 1996, before any claimed priority date.

Neimat was not before the Examiner during prosecution.

C. “Load Management in Distributed Video Servers” by N. Venkatasubramanian (Ex. 1005)

Venkatasubramanian is a secondary reference for Ground 4 and discusses load management techniques. Venkatasubramanian is prior art under pre-AIA §§ 102(a) and (b) because it was a printed publication as of May 1997, at least one year before any claimed priority date.

Venkatasubramanian is an IEEE publication having ISBN number 0-8186-7813-5. (See Ex. 1002, ¶64.) IEEE published this article in the Proceedings of the 17th International Conference on Distributed Computing Systems (ICDCS '97). (*Id.*) According to the attached declaration of Gerard P. Grenier (IEEE Senior Director of Content Management), Venkatasubramanian was made publicly available no later than May 30, 1997. (See Ex. 1036, *see also*, Ex. 1037.)

The Board has previously held similar conference proceedings to be printed publications for the purposes of 35 U.S.C. §§ 102(b) and 311(b). *See Activision Blizzard, Inc. v. Acceleration Bay, LLC*, Case IPR2015-01972, at *18-22 (P.T.A.B. Mar. 23, 2017) (Paper 111) (finding 1997 IEEE conference proceeding publicly

available printed publication); *Ericsson Inc. v. Intellectual Ventures I LLC*, Case IPR2014-00527, at *12 (P.T.A.B. May 18, 2015) (Paper 41) (“Allowing IPR petitioners to rely on the IEEE publication date in an IPR proceeding ... serves the interests of justice.”).

Venkatasubramanian was not before the Examiner during prosecution.

D. Other Evidence of the State of the Art (Ex. 1006-1028)

Exhibits 1006-1028 further reflect the state of the art, level of ordinary skill in the art, common knowledge in the art, and/or common sense in the art, and are therefore also relevant to the background of the invention and the unpatentability analysis herein. (See Ex. 1002, Section V.)

IV. IDENTIFICATION OF CHALLENGE AND RELIEF REQUESTED [37 C.F.R. §§42.104(B)(1)-(2)]

Petitioner requests *inter partes* review of claims 1, 3, 6, 10, 14, 17, 23, 24, 30, and 31 and requests that the Board find these claims to be unpatentable based on the following specific statutory grounds and prior art.

Ground 1: RFC1034 (Ex. 1003) anticipates claims 1, 3, 6, 10, 14, and 31 under 35 U.S.C. § 102(b).

Ground 2: RFC1034 (Ex. 1003) renders obvious claim 6 under 35 U.S.C. § 103(a).

Ground 3: RFC1034 (Ex. 1003) in combination with U.S. Patent No. 5,542,087 to Neimat (Ex. 1004) renders obvious claims 17, 23, and 30 under 35 U.S.C. § 103(a).

Ground 4: RFC1034 (Ex. 1003) in combination with U.S. Patent No. 5,542,087 to Neimat (Ex. 1004) further in view of “Load Management in Distributed Video Servers” by N. Venkatasubramanian (Ex. 1005) renders obvious claim 24 under 35 U.S.C. § 103(a).

Grounds 1-4 are substantially different from the references and arguments considered by the Patent Office during the '978 patent prosecution. During prosecution, RFC1034, Neimat, and Venkatasubramanian were not presented to or cited by the Examiner. Nor was any publication describing DNS in the level of detail of RFC1034. Patent Owner referred generally to DNS in the '978 patent (Ex. 1001, 23:60-24:11); however, AWS relies on much more detailed teachings regarding DNS in this Petition (such as published in RFC1034), along with expert support. Thus, the same or substantially the same prior art or arguments were not previously presented to the Patent Office, and there is no discretionary basis to deny the Petition as redundant.

V. CLAIM CONSTRUCTION [37 C.F.R. § 42.104(B)(3)]

The Board “need only construe terms ‘that are in controversy, and only to the extent necessary to resolve the controversy.’” *Nidec Motor Corp. v.*

Zhongshan Broad Ocean Motor Co., 868 F.3d 1013, 1017 (Fed. Cir. 2017). AWS is presently unaware of any claim construction disputes between the parties relevant to the unpatentability arguments in this Petition. Therefore, AWS does not ask the Board to construe any claim terms at this time.⁴ AWS reserves the right to do so if a claim construction dispute arises.

For purposes of this Petition, AWS applies the ordinary and customary meaning of the claim terms in light of the specification. See 37 C.F.R. §42.100(b); *Phillips v. AWH Corp.*, 415 F.3d 1302, 1312-13 (Fed. Cir. 2005) (en banc). In addition, in several instances, AWS will explain how the relevant prior art discloses each claim limitation even if the claim limitation is construed more narrowly to encompass only specific embodiments or disclosures in the '978 patent specification. Therefore, even if the Board were to adopt claim constructions that are narrower than the ordinary and customary meaning of the claim terms, the same prior art and evidence would render the claims unpatentable.

⁴ Nor does AWS set forth arguments regarding claim construction disputes not properly addressed in this forum, such as those related solely to AWS's defenses based on non-infringement, patent ineligibility, etc.

VI. GROUNDS OF UNPATENTABILITY [37 C.F.R. §§ 42.104(B)(4)-(5)]**A. Ground 1: RFC1034 Anticipates Claims 1, 3, 6, 10, 14, and 31.****1. Overview of Mapping Between '978 Patent and RFC1034**

Patent Owner claims that its patent is inventive because “[w]hen you separate what we characterize as the where from the what, there is some magic in that.” (See Ex. 1041, 26.) But long before the patent, RFC1034 already disclosed a system that separates the “where” from the “what.” In RFC1034, a client accesses a distributed network to determine the address—the “where”—associated with a particular domain name. The client then uses that address to access data—the “what”—associated with that domain name.

The '978 patent claims include the following terms: location information, location servers, entity, identifier, location string, data, location query, location message, and redirect message. Each of these limitations corresponds to a disclosure in RFC1034, as reflected in the chart and explained more fully below.

'978 Claim Limitation	Corresponding Exemplary Disclosure in RFC1034
Location information	Resource records (“RRs”)
Location servers	Name servers
Entity	Host, mailbox, object, or resource
Identifier	Domain name
Location string	Host address
Data	Host data or mailbox data

'978 Claim Limitation	Corresponding Exemplary Disclosure in RFC1034
Location query	Standard query
Location message	Response from name server that answers the question posed in the query
Redirect message	Response from name server that refers the requester to another set of name servers

Location information corresponds to DNS resource records (“RR”). “The set of resource information associated with a particular name is composed of separate resource records (RRs).” (Ex. 1003, 12.) RRs contain information regarding the locations of hosts and mail servers on a network, *i.e.*, “location information.” In particular, they store the host address record associated with a host or the mail server address for a mail exchange associated with a mailbox. (*Id.*, 12; Ex. 1002, ¶72.)⁵

Location server corresponds to DNS name servers. DNS name servers store RRs as well as pointers to other name servers, thus forming a network. The

⁵ The specification states that a “location” is an encoding “that is a member of a set of associations with an identifier in an NDTP server.” (Ex. 1001, 4:28-30.) It further states that “information about where data associated with a particular application entities can be managed and obtained independently of the data itself.” (*Id.*, 23:65-24:1.) RFC1034 contains these disclosures as well. (See, *e.g.*, Ex. 1002, ¶72.)

“name servers” contain “information about the domain tree’s structure and set information” and “complete information about a subset of the domain space, and pointers to other name servers that can be used to lead to information from any part of the domain tree.” (Ex. 1003, 6; Ex. 1002, ¶73.)⁶

Entity corresponds to a DNS host, mailbox, object, or resource. (Ex. 1003, 1, 6, 9, 11, 12.) “[T]he Internet uses some of its domain names to identify hosts.” (*Id.*, 6.) RFC1034 also discloses mailboxes (Ex. 1003, 3; Ex. 1002, ¶74), and states that “hosts” and “mailboxes” are a “normal” part of the name space. (Ex. 1003, 9.) It also refers to “objects” and “resources” in a similar way. DNS “can be used to hold naming information for some kind of object” using “a convention for mapping between object names and domain names.” (*Id.*, 9.) DNS provides “a consistent name space which will be used for referring to resources.” (*Id.*, 2.)

⁶ RFC1034’s name servers also have the same functionality as the ’978 patent’s location servers. As in the ’978 patent, each DNS name server maintains a set of domain name/address mappings that are modified or returned in response to DNS protocol request messages from clients. (See, e.g., Ex. 1002, ¶73; Ex. 1001, 4:34-38.) Like the transfer protocol described in the ’978 patent, the DNS transfer protocol has the capability to manipulate location information associated with an individual entity in the distributed database system. (See, e.g., Ex. 1002, ¶¶138-141; Ex. 1001, 4:20-25.)

Identifier corresponds to DNS domain names. Hosts and mailboxes are identified by identifiers—their domain names. (Ex. 1003, 1, 6, 7, 8; Ex. 1002, ¶75.) For example, venera.isi.edu is such a domain name for a host.

For example, we might show the RRs carried in a message as:

ISI.EDU.	MX	10 VENERA.ISI.EDU.
	MX	10 VAXA.ISI.EDU.
VENERA.ISI.EDU.	A	128.9.0.32
	A	10.1.0.52
VAXA.ISI.EDU.	A	10.2.0.27
	A	128.9.0.33

(Ex. 1003, 14 (emphasis added); Ex. 1002, ¶¶75, 88.)

Email addresses are “mapped into a domain name” that identifies the host responsible for the mailbox. (Ex. 1003, 9; Ex. 1002, ¶¶75, 76, 87, 95.) In the above example, venera.isi.edu is the domain name associated with email addresses ending in “isi.edu.”⁷

Location string corresponds to DNS host addresses. (Ex. 1003, 1, 3, 6, 9; Ex. 1002, ¶76.) In the above example, the host addresses associated with the domain name venera.isi.edu are 128.9.0.32 and 10.1.0.52.

Data corresponds to DNS host data or mailbox data. In RFC1034, the IP address of an entity specifies a location of data corresponding to that entity. The

⁷ The specification recites that identifiers are unique encodings, such as strings, with which zero or more data location specifiers are associated in an NDTP server. (Ex. 1001, 4:25-28.) RFC1034 contains this disclosure. (Ex. 1002, ¶75.)

client can then retrieve the data at that IP address. For example, a client can retrieve “mailbox data” associated with a particular mailbox. (Ex. 1003, 3; Ex. 1002, ¶¶77-78.)

Location query corresponds to DNS standard queries. (Ex. 1003, 22). “A query names the domain name of interest and describes the type of resource information that is desired.” (Ex. 1003, 6; Ex. 1002, ¶79.)

Location message and **redirect message** correspond to two different types of responses from DNS name servers. “The response by the name server either answers the question posed in the query, refers the requester to another set of name servers, or signals some error condition.” (Ex. 1003, 16; Ex. 1002, ¶¶80-81.) **Location message** corresponds to the response that “answers the question posed in the query”; **redirect message** corresponds to the response that “refers the requester to another set of name servers.”

For completeness, Dr. Weissman’s declaration also includes claim charts further reflecting how these claim limitations and all others are met by the prior art. (Ex. 1002, Appendix B.)

2. Independent Claim 1

RFC1034 discloses each limitation in claim 1.

RFC1034 discloses **[1.Preamble]**: “[a] system having a plurality of location servers for managing location information and providing location information to

location queries.” As described above, RFC1034 discloses a “plurality of location servers”—*i.e.*, name servers. (Ex. 1002, ¶¶73, 83.) These name servers manage “location information”—*i.e.*, the RRs that are stored on the name servers. (*Id.*, ¶¶72, 83)

The name servers also provide location information in response to “location queries.” “The principal activity of name servers is to answer standard queries.” (Ex. 1003, 22.) “A query names the domain name of interest and describes the type of resource information that is desired.” (*Id.*, 6.)

RFC1034 discloses [1.a.1]: “*a first location server containing a first set of location information corresponding to at least one entity.*” As described above, RFC1034 discloses a “first location server”—*i.e.*, a first name server. RFC1034 also discloses a “first set of location information”—*i.e.*, an RR (or RRs) that contain host addresses associated with particular domain names. Finally, the location information “correspond[s] to at least one entity”: the corresponding host or mailbox, for example. (*Supra*, at VI.A.1; Ex. 1002, ¶¶84-88.)

RFC1034 discloses [1.a.2]: “*the location information comprising an identifier and at least one location string associated with the identifier.*” In RFC1034, the “location information” (RR) comprises an “identifier” (domain name) and “at least one location string associated with the identifier” (address associated with the domain name). (*Supra*, at VI.A.1; Ex. 1002, ¶¶89-92.)

RFC1034 discloses **[1.a.3]**: “*wherein the identifier identifies an entity and the location string specifies a location of data pertaining to the entity.*” In RFC1034, the “identifier” (domain name) identifies an “entity” (host or mailbox) and the “location string” (address) specifies a “location of data pertaining to the entity” (location of data pertaining to the host or mailbox, such as mail data). (*Supra*, at VI.A.1; Ex. 1002, ¶¶93-99.)

RFC1034 discloses **[1.b]**: “*a second location server comprising a second set of location information, wherein at least a portion of the second set of location information differs from the first set of location information.*” RFC1034 describes a first name server, such as discussed above for elements **[1.a.1-3]**. In addition, RFC1034 discloses multiple “name servers” and that a first one of these name servers must point to at least a second “other server.” (Ex. 1003, 4, 6; Ex. 1002, ¶100.)

RFC1034 contains several disclosures demonstrating that in the “other server,” “at least a portion of the second set of location information differs from the first set of location information,” as required by the claim. First, RFC1034 discloses: “[I]n general a particular name server *has complete information about a subset* of the domain space, and pointers to other name servers that can be used to lead to information from any part of the domain tree.” (Ex. 1003, 6 (emphasis added).) Because one name server has information about a subset of the domain

space, and other name servers have information about other subsets of the domain space, RFC1034 discloses multiple location servers with different location information. (Ex. 1002, ¶¶101-102.)

Second, RFC1034 discloses that the database “is divided up into sections called zones, which are distributed among the name servers.” (Ex. 1003, 18.) A first name server has authoritative location information about “only a small section of the domain tree.” (*Id.*, 19.) A second name server, therefore, has different authoritative information about another section. One of ordinary skill would have understood that location information necessarily differs among a first server in a first zone and a second server in a second zone. (Ex. 1002, ¶¶102-103, 105.)

Third, RFC1034 describes a redirect capability that requires at least a first and second name server with different location information. (*See* Ex. 1003, 16.) The purpose of the redirect is to refer the requester to another set of name servers if the first name server does not have the requested information. (Ex. 1002, ¶104.) If all DNS servers had the same information, there would be no reason to redirect. (*Id.*)

Therefore, RFC1034 discloses a first name server and second name server containing different location information.

RFC1034 discloses [1.c.1]: “*programming logic stored on each of the location servers responsive to a location query identifying a desired entity to return a*

location message.” RFC1034 refers to “DNS software,” “server programs,” “binary” packets, and “algorithm[s],” that provide programming logic. (Ex. 1003, 6, 13, 21, 24; Ex. 1002, ¶107.) Also, the name servers necessarily include programming logic to carry out such computer functions. (Ex. 1002, ¶¶111-112.) RFC1034 states the name servers are responsive to location queries identifying a desired resource to return a location message: “Queries are messages which may be sent to a name server to provoke a response.... The response by the name server either answers the question posed in the query, refers the requester to another set of name servers, or signals some error condition.” (Ex. 1003, 15-16; Ex. 1002, ¶¶108-110.)

RFC1034 discloses [1.c.2]: “*the location message comprising at least one location of data pertaining to the desired entity, if the location server receiving the location query contains location information for the desired entity.*” RFC1034 discloses messages from DNS name servers that satisfy this limitation. “Using the query domain name, ... the name server looks for matching RRs” to return a corresponding location message. (Ex. 1003, 17.) For example, a mailer trying to send mail to Mockapetris@ISI.EDU might ask the resolver for location information about ISI.EDU. The response’s “location message” would be:

```

ISI.EDU.      MX      10 VENERA.ISI.EDU.
               MX      10 VAXA.ISI.EDU.

```

while the additional section might be:

```

VAXA.ISI.EDU.  A      10.2.0.27
               A      128.9.0.33
VENERA.ISI.EDU. A      10.1.0.52
               A      128.9.0.32

```

(*Id.* 17.) The “additional section” is provided “[b]ecause the server assumes that if the requester wants mail exchange information, it will probably want the addresses of the mail exchanges soon afterward.” (*Id.*)

These messages comprise “at least one location of data pertaining to the desired entity”—*i.e.*, the location of mailbox data pertaining to a desired mailbox. They are sent “if the location server receiving the location query contains location information for the desired entity”—*i.e.*, if the name server has the correct RR. (Ex. 1002, ¶¶116-117.)

Below is a claim chart for claim 1:

U.S. Patent No. 7,233,978	Exemplary Disclosures of RFC1034
[1.Preamble] A system having a plurality of location servers for managing location information and providing location information to location queries, the system compris-	“Name servers are server programs which hold information about the domain tree's structure and set information. A name server may cache structure or set information about any part of the domain tree, but in general a particular name server has complete information about a subset of the domain space, and pointers to other name servers that can be used to lead to information from any part of the domain tree.” (Ex. 1003, 6.) “A domain name identifies a node. Each node has a set of resource information, which may be empty. The set of resource information associated with a particular name is composed of

ing:	separate resource records (RRs).” (<i>Id.</i>, 11-12.) “The principal activity of name servers is to answer standard queries.” (<i>Id.</i>, 22.) “[Q]uery operations are attempts to extract specific types of information from a particular set. A query names the domain name of interest and describes the type of resource information that is desired. For example, the Internet uses some of its domain names to identify hosts; queries for address resources return Internet host addresses.” (<i>Id.</i>, 6.) In response to a query, a “name server either answers the question posed in the query, refers the requester to another set of name servers, or signals some error condition.” (<i>Id.</i>, 16.)
[1.a.1] a first location server containing a first set of location information corresponding to at least one entity,	See above for disclosures of “location server” and “location information.” “[I]n general a particular name server has complete information about a subset of the domain space, and pointers to other name servers that can be used to lead to information from any part of the domain tree.” (<i>Id.</i>, 6) The information in a name server corresponds to a “resource” such as a “host,” “alias,” “CPU and OS,” or “a mail exchange.” (<i>Id.</i>, 12.) RFC1034 discloses hosts (<i>id.</i>, 1, 6, 9, 11, 12) and mailboxes (<i>id.</i>, 3, 4, 9). RFC1034 states “hosts” and “mailboxes” are a “normal” part of the name space. (<i>Id.</i>, 9.) It also refers to “objects” and “resources” in a similar way. “DNS can be used to hold naming information for some kind of object” using “a convention for mapping between object names and domain names.” (<i>Id.</i>, 9.) The “primary goal” of DNS “is a consistent name space which will be used for referring to resources.” (<i>Id.</i>, 2.) A domain name is represented by a “dotted quad” in symbolic form (e.g., SRI-NIC.ARPA) and associated address of the domain name is a “dotted quad” in numeric form, the so-

	called IP-address (e.g., 26.0.0.73). This example is shown below from RFC1034. <pre> SRI-NIC.ARPA. A 26.0.0.73 A 10.0.0.51 MX 0 SRI-NIC.ARPA. HINFO DEC-2060 TOPS20 </pre> (<i>Id.</i>, 37 (emphasis added).)
[1.a.2] the location information comprising an identifier and at least one location string associated with the identifier,	A “domain name identifies a node.” (<i>Id.</i>, 11.) “Conceptually, each node and leaf of the domain name space tree names a set of information, and query operations are attempts to extract specific types of information from a particular set. A query names the domain name of interest and describes the type of resource information that is desired. For example, the Internet uses some of its domain names to identify hosts; queries for address resources return Internet host addresses.” (<i>Id.</i>, 6; <i>see also id.</i>, 12.) For example, the identifier “vaxa.isi.edu” has two IP addresses shown below. For example, we might show the RRs carried in a message as: <pre> ISI.EDU. MX 10 VENERA.ISI.EDU. MX 10 VAXA.ISI.EDU. VENERA.ISI.EDU. A 128.9.0.32 A 10.1.0.52 VAXA.ISI.EDU. A 10.2.0.27 A 128.9.0.33 </pre> (<i>Id.</i>, 14, (emphasis added).)
[1.a.3] wherein the identifier identifies an entity and the location string specifies a location of data pertaining to the entity	The name space is “used for hosts, mailboxes, etc.” (<i>Id.</i>, 9). The “mailbox HOSTMASTER@SRI-NIC.ARPA is represented as a domain name by HOSTMASTER.SRI-NIC.ARPA.” (<i>Id.</i>, 9-10). A client can use DNS to retrieve “mailbox data” associated with a particular mailbox. (<i>Id.</i>, 3).
[1.b] a second location server comprising a second set of loca-	“A particular name server has complete information about a subset of the domain space, and pointers to other name servers that can be used to lead to information from any part of the domain tree.” (<i>Id.</i>, 6.)

tion information, wherein at least a portion of the second set of location information differs from the first set of location information; and	“A given name server will typically support one or more zones, but this gives it authoritative information about only a small section of the domain tree.” (<i>Id.</i>, 19.) “In any system that has a distributed database, a particular name server may be presented with a query that can only be answered by some other server.” (<i>Id.</i>, 4.)
[1.c.1] programming logic stored on each of the location servers responsive to a location query identifying a desired entity to return a location message,	See analysis of preamble for disclosures of “location servers responsive to a location query identifying a desired entity.” “Queries are messages which may be sent to a name server to provoke a response. ... The response by the name server either answers the question posed in the query, refers the requester to another set of name servers, or signals some error condition.” (<i>Id.</i>, 15-16.) RFC1034 discloses that name servers are “server programs” (<i>id.</i>, 6) that respond to queries with a DNS response answer. RFC1034 discloses that DNS name servers “answer standard queries” and that they send “response[s]” that “are carried in a standard message format.” (<i>Id.</i>, 22.) DNS name servers contain “DNS software” that is run when a name server is locating information contained in a RR responsive to a location query. (<i>Id.</i>, 15.) “The actual algorithm used by the name server will depend on the local OS and data structures used to store RRs.” (<i>Id.</i>, 24.)
[1.c.2] the location message comprising at least one location of data pertaining to the desired entity, if the location server receiving the loca-	“Using the query domain name, QTYPE, and QCLASS, the name server looks for matching RRs.” (<i>Id.</i>, 17.) For example, a mailer trying to send mail to Mockapetris@ISI.EDU might ask the resolver for mail information about ISI.EDU, resulting in a query for QNAME=ISI.EDU, QTYP=MX, QCLASS=IN. The response’s answer section would be:”

tion query contains location information for the desired entity.	ISI.EDU.	MX MX	10 VENERA.ISI.EDU. 10 VAXA.ISI.EDU.
	while the additional section might be:		
	VAXA.ISI.EDU.	A A	10.2.0.27 128.9.0.33
	VENERA.ISI.EDU.	A A	10.1.0.52 128.9.0.32
	(<i>Id.</i>, 17.) The additional section is provided “[b]ecause the server assumes that if the requester wants mail exchange information, it will probably want the addresses of the mail exchanges soon afterwards.” (<i>Id.</i>) “A standard query specifies a target domain name (QNAME), query type (QTYPE), and query class (QCLASS) and asks for RRs which match.” (<i>Id.</i>, 16.) See also Section 6.2 providing example queries and responses. (<i>Id.</i>, 39-50.)		

3. Dependent Claims 3 and 6

Claim 3 recites: “*The system of claim 1, wherein the programming logic further comprises logic responsive to the location query to return one of a location message or a redirect message, wherein the location server receiving the query returns the location message if the queried location server contains location information for the desired entity, and wherein the queried location server returns a redirect message if the queried location server lacks location information for the desired entity, the redirect message comprising information for finding a location server known to have location information relevant to the location query.*”

RFC1034 anticipates claim 3. Name servers may return either a “location message” (if the location appears on the server) or a “redirect message” (if the location does not appear on the server), as required by the claim: “Queries are messages which may be sent to a name server to provoke a response.... The response by the name server either answers the question posed in the query, refers the requester to another set of name servers, or signals some error condition.” (Ex. 1003, 15-16.)

In RFC1034, the redirect message comprises “information for finding a location server known to have location information relevant to the location query,” as required by the claim: “In addition to relevant records, the name server may return RRs that point toward a name server that has the desired information or RRs that are expected to be useful in interpreting the relevant RRs. For example, a name server that doesn’t have the requested information may know a name server that does; a name server that returns a domain name in a relevant RR may also return the RR that binds that domain name to an address.” (Ex. 1003, 17; Ex. 1002, ¶¶121-123.)

Claim 6 recites: “*The system of claim 1, wherein the location information in the location server is maintained in an indexed location store indexed by a hash table.*” RFC1034 discloses that “many name servers internally form tree or hash structures for the name space.” (Ex. 1003, 13.)

One of ordinary skill would have understood that the term “hash structure” in RFC1034 is synonymous with “hash table.” A “hash structure” is simply a data structure that maps hash indices to items in the data structure—*i.e.*, a hash table. (Ex. 1002, ¶131.)

One of ordinary skill would have further understood that RFC1034’s disclosure of a “hash structure for the name space” inherently discloses that “the location information in the location server is maintained in an indexed location store indexed by a hash table,” as required by the claim. One of ordinary skill would have understood that because resource records (*i.e.*, location information) are stored on the name servers, the name servers contain a “location store.” (*Id.*, ¶126.) One of ordinary skill would have also understood that a hash structure (hash table) forms an index by applying a hash function to the domain name to generate an index value within the range of the hash table data structure. (*Id.*, ¶133.) Thus, one of ordinary skill would have understood that the DNS name server uses this internal hashing structure to form an index to the indexed location store, satisfying the claim limitation. (*Id.*, ¶134.)

4. Independent Claim 10

Independent claim 10 contains many of the same limitations as representative independent claim 1 discussed above (*e.g.*, location server, location infor-

mation, identifier, location message, etc.). The analysis of those limitations applies equally to claim 10.

RFC1034 discloses **[10.a.1]**: “a plurality of location servers containing location information corresponding to a plurality of entities, the location information formatted according to a transfer protocol configured for manipulating location information.” As explained above, RFC1034 discloses “a plurality of location servers containing location information corresponding to a plurality of entities.”

RFC1034 also discloses the claimed “transfer protocol configured for manipulating location information”: “This RFC introduces domain style names, their use for Internet mail and host address support, and the protocols used to implement domain name facilities.” (Ex. 1003, 1.) “A subset of DNS functions and data types constitute an official protocol. The official protocol includes standard queries and their responses and most of the Internet class data formats (e.g., host addresses).” (*Id.*) Because the “standard queries and their responses” are queries for location information, RFC1034 discloses a “transfer protocol configured for manipulating location information.” (Ex. 1002, ¶141.)

RFC1034 discloses the transfer of location information from one location server to another. “Whole zones can be transferred between name servers by transferring the RRs, either carried in a series of messages or by FTPing a master file which is a textual representation.” (Ex. 1003, 20.) This transfer takes place ac-

cording to the “zone transfer part of the DNS protocol,” thus satisfying the limitation that the “*transfer protocol*” is “*configured for manipulating location information.*” (Ex. 1003, 28; *see* Ex. 1002, ¶¶138-141.)

RFC1034 discloses that the location information in RFC1034 is “*formatted according to*” the transfer protocol: “RRs are represented in binary form in the packets of the DNS protocol, and are usually represented in highly encoded form when stored in a name server or resolver.” (Ex. 1003, 13.)

RFC1034 discloses [10.a.2]: “*comprising at least one application server address.*” The application server address in RFC1034 may be, *e.g.*, a mail server address that is part of a RR. (*Supra*, at VI.A.1; Ex. 1003, 13, 17.) One of ordinary skill would have understood that a mail server is a type of application server. (Ex. 1002, ¶142.)

RFC1034 discloses [10.a.3]: “*wherein the plurality of location servers are arranged in a cluster topology such that each location server contains a unique set of location information of an aggregate set of the location information.*” The ’978 patent describes a server cluster as the set of machines managing the server data set (*i.e.*, identifier and location information): “The set of machines managing the NDTP server data set may be referred to as an NDTP server cluster 50, as shown in FIG. 5.” (Ex. 1001, 15:50-52.) Figure 5 of the ’978 patent depicts that the servers in the cluster are interconnected to enable redirection of query requests, given that

different servers contain different identifier mappings. (Ex. 1001, Fig. 5; Ex. 1002, ¶145.)

Similarly, RFC1034 discloses that the DNS name space is partitioned into zones, such as “by ‘class’ and by ‘cuts’ made in the name space between nodes.” (Ex. 1003, 19.) DNS name servers manage one or more zones based on the partitioning on the DNS name space and serve queries for a unique set of data contained in their zone: “The database is divided up into sections called zones, which are distributed about the name servers.... [T]he essential task of a name server is to answer queries using data in its zones.” (*Id.*, 18.)

Because the DNS database is partitioned and queries may be submitted to any name server, the DNS name servers are interconnected to enable a query to be redirected to the name server holding the unique identifier/location mapping for a query. (Ex. 1003, 4; Ex. 1002, ¶146.) One of ordinary skill would have understood that this constitutes a “cluster topology such that each location server contains a unique set of location information of an aggregate set of the location information.” (Ex. 1002, ¶147.)

RFC1034 discloses **[10.b]**: “*programming logic stored on each of the plurality of location servers responsive to a location query for a desired identifier to return one of a location message, wherein a queried location server returns a location message if the queried location server contains location information for the*

desired identifier, and a redirect message if the queried location server does not contain location information relevant to the desired identifier, wherein the redirect message comprises information for finding a location server having location information related to the desired identifier.” This limitation is closely similar to the limitation in dependent claim 3, addressed above. According to RFC1034, if a queried name server does not contain an RR relevant to the desired domain name, the name server sends a response message to “refer[] the requester to another set of name servers.” (Ex. 1003, 15-16.) “[T]he name server may return RRs that point toward a name server that has the desired information or RRs that are expected to be useful in interpreting the relevant RRs.” (*Id.*, 17.)

5. Independent Claim 14

Independent claim 14 contains many of the same limitations as representative independent claim 1 (*e.g.*, location server, location information, identifier, location message, etc.) and independent claim 10 (*e.g.*, redirect message). The analysis of those limitations applies equally to claim 14.

RFC1034 discloses **[14.preamble]**: “*A method of handling location queries in a network, the network comprising a plurality of location servers, each location server containing a unique set of location information of an aggregate set of location information correlating each of a plurality of identifiers with at least one location.*” For the reasons explained with respect to claim 1, RFC1034 discloses a

method of handling location queries in a network comprising a plurality of location servers. (*Supra*, at VI.A.2.) For the reasons stated with respect to claim 10, RFC1034 discloses “each location server containing a unique set of location information of an aggregate set of location information correlating each of a plurality of identifiers with at least one location.” (*Supra*, at VI.A.4.)

RFC1034 discloses **[14.a]** “*receiving a location query from a client at one of the plurality of location servers, the location query requesting an entity’s location.*”⁸ RFC1034 discloses that a client sends a location query: “Clients of the domain system should be able to identify trusted name servers they prefer to use.” (*Id.* at 4.) For the reasons stated above with respect to claim 1, RFC1034 discloses every other limitation in **[14.a]**. (*Supra*, at VI.A.2.)

RFC1034 discloses **[14.b]**: “*sending a location response message to the client if the queried location server contains location information relevant to an entity identified in the query, the location response message comprising location in-*

⁸ As in the ’978 patent, a client communicates with DNS name servers to obtain location information. (See, e.g., Ex. 1002, ¶¶40, 152-153; Ex. 1001, 4:30-38.) In RFC1034, for example, the client is a network attached component that initiates add, delete, lookup, and update of domain name/host address mappings and associations from a name server with standard queries. (*Id.*)

formation identifying at least one application server containing information relevant to the entity identified in the query.” Each of these limitations appears in claim 1, except for the “application server” limitation which appears in claim 10. For the reasons stated above, RFC1034 discloses all these limitations. (*Supra*, at VI.A.4.)

RFC1034 discloses **[14.c]**: “*sending a redirect message to the client if the queried location server does not contain data location information relevant to the entity identified in the query, the redirect message comprising information for finding a location server storing the entity identified in the query.*” For the reasons stated above with respect to claim 3, RFC1034 discloses the claimed redirect message. (*Supra*, at VI.A.3.) RFC1034 also discloses sending that redirect message to the *client*. It notes the “iterative” approach “in which the server refers the client to another server and lets the client pursue the query,” and states that the “domain system requires implementation of the iterative approach.” (Ex. 1003, 4.)

6. Independent Claim 31

Independent claim 31 contains the same limitations as claims previously addressed (*e.g.*, location server, location information, identifier). The claim chart below explains how each claim limitation is substantively identical to claim limitations that have previously been addressed:

U.S. Patent No. 7,233,978	Corresponding Claim Limitations
[31.Preamble] A system for managing location information and providing location information to location queries, the system comprising:	See [1.Preamble]; Ex. 1002, ¶159.
[31.a] a location server operating in accordance with a transfer protocol, the transfer protocol comprising instructions for manipulating an identifier and at least one location associated with the identifier, wherein the identifier uniquely specifies an entity and wherein each location specifies a location of data in a network pertaining to the entity, the location server containing location information corresponding to at least one entity and formatted according to the transfer protocol, and wherein the location of data for the location comprises an application server in communication with the network; and	See [10.a.1] and [10.a.2]; Ex. 1002, ¶160.
[31.b] programming logic stored on the location server responsive to a location query identifying a desired entity to return a location message, the location message comprising locations associated with the desired entity, wherein the location server returns the location message if the location server contains location information for the desired entity.	See [1.c.1] and [1.c.2]; Ex. 1002, ¶161-162.

B. Ground 2: RFC1034 Renders Obvious Claim 6.

It is a “fundamental proposition that obvious variants of prior art references are themselves part of the public domain.” *In re Translogic Tech., Inc.*, 504 F.3d 1249, 1259 (Fed. Cir. 2007) (citing *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398 (2007)). “A patent can be obvious in light of a single prior art reference if it would

have been obvious to modify that reference to arrive at the patented invention.” *Game & Tech. Co. v. Activision Blizzard Inc.*, 926 F.3d 1370, 1381 (Fed. Cir. 2019) (quotation marks omitted). Variants that are only minimally different from the prior art are especially likely to be obvious. *See, e.g., ZUP, LLC v. Nash Mfg., Inc.*, 896 F.3d 1365, 1374 (Fed. Cir. 2018) (finding obviousness “where the differences between the claimed invention and the prior art are minimal”); *Kimberly-Clark Worldwide, Inc. v. First Quality Baby Prods., LLC*, 431 F. App’x 884, 888 (Fed. Cir. 2011) (holding that where “the practical difference between using vacuum rolls instead of vacuum conveyors appears very minimal,” vacuum conveyors were “an obvious variant of the vacuum rolls”).

The Patent Office has previously held that a challenger can argue that a claim was anticipated by prior art, and, in the alternative, so closely similar to the prior art as to be obvious in light of that prior art. *See, e.g., SZ DJI Tech. Co. v. Synergy Drone LLC*, Case IPR2018-00205, at *21 (P.T.A.B. May 23, 2019) (Paper 47) (“Moreover, we are persuaded by Petitioner’s alternative argument that even if Thornberg-983 were deemed to not expressly disclose radio control signals, it would have been obvious to utilize Thornberg-983’s control panel 200 to transmit RF signals to the vehicle because an ordinarily skilled artisan would have recognized that such signals provide the range appropriate for a remotely controlled aircraft.”).

Here, too, even if there is some argument that claim 6 is not anticipated by RFC1034, any differences would be so trivial that “it would have been obvious to modify that reference to arrive at the patented invention.” *Game & Tech*, 926 F.3d at 1381.

Claim 6 depends from claim 1 and includes the following limitation: “*the location information in the location server is maintained in an indexed location store indexed by a hash table.*” RFC1034 discloses that “many name servers internally form tree or hash structures for the name space.” (Ex. 1003, 13.) Even if there is some minor conceivable distinction between RFC1034’s disclosure and claim 6, RFC1034 would render claim 6 obvious. A “hash table” would be obvious to an ordinarily skilled artisan based on the disclosure of a “hash structure.” (Ex. 1002, ¶131.) An ordinarily skilled artisan looking at RFC1034 would also have understood that a hash structure forms an index by applying a hash function to the input (*i.e.*, a domain name from the “name space”) to generate an index value within the range of the hash table data structure. (*Id.*, ¶133.) The ordinarily skilled artisan would have similarly understood that because the point of the hash function is to allow the retrieval of location information corresponding to a domain name, the location information would be indexed according to the hash function and stored in a location store. (*Id.*, ¶134.)

Thus, at a minimum, it would have been obvious from RFC1034's disclosure of the "hash structure for the name space" that "the location information in the location server is maintained in an indexed location store indexed by a hash table" as in claim 6. (*Id.*, ¶135.)

C. Ground 3: RFC1034 in view of Neimat Renders Obvious Claims 17, 23, and 30.

1. RFC1034 and Neimat, in combination, disclose all limitations in Independent Claim 17 and Dependent Claims 23 and 30.

RFC1034 (Ex. 1003) in view of Neimat (Ex. 1004) renders obvious claims 17, 23, and 30. RFC1034 discloses all elements of claim 17 except for element [17.e]: "*transferring a portion of the identifiers and associated locations to a second data location server when a performance criterion of the first location server reaches a predetermined performance limit.*" Neimat renders obvious this limitation and the claim as a whole, as well as the subject matter in dependent claims 23 and 30. As explained below, one of ordinary skill would have been motivated to combine RFC1034 and Neimat with a reasonable expectation of success.

RFC1034 discloses [17.Preamble]: "*[a] method of scaling at least one of capacity and transaction rate capability in a location server in a system having a plurality of location servers for storing and retrieving location information, wherein each of the plurality of location servers stores unique set of location information of an aggregate set of location information.*"

RFC1034 discloses methods of scaling capacity and transaction rate capability. RFC1034 discloses that in view of the “sheer size of the database,” “it must be maintained in a distributed manner, with local caching to improve performance.” (Ex. 1003, 2; Ex. 1002, ¶169.) One of ordinary skill would have understood this to be a disclosure of a “method of scaling capacity capability”—*i.e.*, adding servers to a distributed network to account for the “sheer size of the database.” (Ex. 1002, ¶170.) One of ordinary skill would have also understood that “local caching” ensures that by adding local resources to a network—which allows data to be cached nearby—the network’s transaction rate improves. (*Id.*) Thus, one of ordinary skill would have understood that RFC1034 discloses a method of scaling transaction rate capability. (*Id.*)

Moreover, the idea of a “method of scaling at least one of capacity and transaction rate capability” would have been obvious, as the goal of making systems faster and bigger is well-known. See *DyStar Textilfarben GmbH v. C.H. Patrick Co.*, 464 F.3d 1356, 1368 (Fed. Cir. 2006) (“Because the desire to enhance commercial opportunities by improving a product or process is universal—and even common-sensical—we have held that there exists in these situations a motivation to combine prior art references even absent any hint of suggestion in the references themselves.”).

As explained below, RFC1034 does not disclose the particular method of scaling capacity or transaction rate capability recited later in the claim, but Neimat, renders this limitation and the claim as a whole obvious. (*See infra*, at [17.e].)

RFC1034 discloses **[17.a]**: “*providing a transfer protocol configured to transport identifier and location information, the location information specifying the location of information related to the identifier,*” as well as **[17.b]** “*storing location information formatted according to the transfer protocol at a first location server.*” As explained above, RFC1034 discloses **[10.a.1]**, which requires both a “transfer protocol configured for manipulating location information” and the “location information formatted according to a transfer protocol.” (*Supra*, at VI.A.4.) RFC1034 also discloses that “the location information specif[ies] the location of information related to the identifier.” (*Supra*, at VI.A.1.)

RFC1034 discloses **[17.c]**: “*receiving an identifier and a location relevant to the identifier at the first location server.*” In RFC1034, name servers receive the claimed identifier (DNS “domain names”) and the claimed location (DNS “host address”). (*See* Ex. 1003, 12.) To “store” RRs in a name server, a name server must receive this information. (Ex. 1002, ¶¶188-189.) A name server may receive this information from the DNS resolver. (*Id.*, ¶68.)

RFC1034 discloses **[17.d]**: “*storing the received location in a location store at the first data location server, the location store comprising a plurality of identi-*

fiers, each identifier associated with at least one location, wherein the received location is associated with the received identifier in the location store.” As discussed above, RFC1034 describes that it “stores” RRs, including the claimed received location, in name servers. “RRs are represented in binary form in the packets of the DNS protocol, and are usually represented in highly encoded form when stored in a name server.” (Ex. 1003, 13; *see also id.*, 24.) The portion of the name server that stores RRs meets the location store limitation. (Ex. 1002, ¶191.)

The example below shows that, in an RR, there may be “a plurality of identifiers, each identifier associated with at least one location, wherein the received location is associated with the received identifier in the location store,” as required by the claim:

For example, we might show the RRs carried in a message as:

ISI.EDU.	MX	10 VENERA.ISI.EDU.
	MX	10 VAXA.ISI.EDU.
VENERA.ISI.EDU.	A	128.9.0.32
	A	10.1.0.52
VAXA.ISI.EDU.	A	10.2.0.27
	A	128.9.0.33

(Ex. 1003, 14; Ex. 1002, ¶185.)

RFC1034 discloses the first portion of [17.e] “*transferring a portion of the identifiers and associated locations to a second data location server.*” Whole zones of resource records are transferred between name servers in RFC1034:

“Whole zones can be transferred between name servers by transferring the RRs, either carried in a series of messages or by FTPing a master file which is a textual representation.” (Ex. 1003, 20.) When changes to a zone are made, they are distributed via such zone transfers: “[T]he preferred method [of distribution] is the zone transfer part of the DNS protocol.” (*Id.*, 28.)

One of ordinary skill would have understood that these “zone transfers” comprise the transfer of “a set of RRs,” which satisfy the requirement of “transferring a portion of the identifiers and associated locations to a second data location server.” (Ex. 1002, ¶194.) At a minimum, in view of these disclosures, it would have been obvious to an ordinarily skilled artisan to “transfer[] a portion of the identifiers and associated locations to a second data location server.” (*Id.*, ¶189, 194.)

RFC1034 does not expressly disclose the second portion of [17.e]: “*when a performance criterion of the first location server reaches a predetermined performance limit.*” However, as shown below, the combination of RFC1034 and Neimat discloses all of [17.e] and the claim as a whole.

In general, Neimat is directed to a distributed computing system that ensures that “clients and servers are able to locate and retrieve a particular data record as accurately and quickly as possible.” (Ex. 1004, 8:16-27; Ex. 1002, ¶174.) In Neimat, “each data record comprises two sections: a short record key identifying

the record, and a remainder of information associated with that key.” (Ex. 1004, 8:27-30; Ex. 1002, ¶196.) These data records are contained in “buckets,” which are memory locations on servers in the distributed network. (See, *e.g.*, Ex. 1004, Abstract, 2:29-30, 4:53-54, 5:30-42; Ex. 1002, ¶197.)

As shown below, record keys 216, 321, 32, and 10 are stored at LOC 0 at SERVER 0:

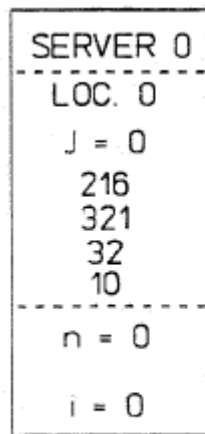


FIG. 3B

(Ex. 1004, Fig. 3B; Ex. 1002, ¶201.)

Neimat teaches a “predetermined performance limit.” Specifically, that predetermined performance limit is reached when “memory location 0 becomes full,” in which case “what is known as a ‘collision’ occurs.” (Ex. 1004, 13:26-27; Ex. 1002, ¶¶201-202.) One of ordinary skill would understand that the “predetermined performance limit” here is that a memory location becomes full.

This analysis is consistent with the '978 patent specification, which characterizes a storage location becoming full as a type of “predetermined performance limit.” It states that “an NDTP server” may “need[] to transfer some of its data due to performance limitations (***physical storage capacity limits***, transaction rate limitations, etc.).” (Ex. 1001, 21:37-41 (emphasis added).) Claim 23, which depends from claim 17, similarly characterizes “an amount of available persistent storage space” as a performance criterion that reaches a “predetermined performance limit.”

Neimat discloses “transferring a portion” of the “identifiers” and associated “information” to a “second data location server.” Specifically, Neimat describes a process known as “splitting” one of the memory locations or buckets, and transferring records to newly added memory: “In response to each collision, the respective predetermined memory location or ‘bucket’ is ‘split’ by creating a respective new memory location, and moving approximately half of the records from the respective predetermined memory location to the respective new memory location.” (Ex. 1004, 13:32-37; Ex. 1002, ¶200.) As noted above, the “records” that are transferred consist of “identifiers” and associated “information”: “each data record comprises two sections: a short record key identifying the record, and a remainder of information associated with that key.” (Ex. 1004, 8:27-30; Ex. 1002, ¶196.) This disclosure of the transfer of records in response to a memory location becom-

ing full also appears in the '978 patent specification, which states that when “an NDTP server needs to transfer some of its data due to performance limitations” such as “physical storage capacity limits,” “copying a subset of data from a first server to a second server ... proceeds according to the splitting process.” (Ex. 1001, 21:37-45.)

Consistent with the claim preamble, Neimat also discloses a method of scaling using that “splitting” process: “When the total number of memory locations, T , have been ‘split’, the table size is doubled to a new total number of memory locations, $2T$... In accordance with such linear expansion principles, the table continues expanding as long as required.” (Ex. 1004, 3:34-46; Ex. 1002, ¶176.)

The result of Neimat’s splitting process is depicted in Neimat’s Figure 3C, as an identifier corresponding to record key value “321” from Figure 3B has been transferred from server 0 (LOC 0) to server 1 (LOC 1). Below, the server portions of Figures 3B and 3C are shown next to each other to show this transfer.

SERVER 0	
LOC. 0	
J = 0	
216	
321	
32	
10	

n = 0	
i = 0	

FIG. 3B

SERVER 0	
LOC. 0	
J = 1	
216	
32	
10	

n = 0	
i = 1	

SERVER 1	
LOC. 1	
J = 1	
153	
321	

FIG. 3C

(Ex. 1002, ¶¶201-202.)⁹

Thus, the combination of RFC1034 and Neimat teach every limitation of [17.e]. Neimat discloses the transfer of identifiers and associated information from one server to a different server when the performance criterion at the first server reaches a predetermined performance limit. (*Id.*, ¶206.) RFC1034 teaches the storage of identifiers and location information at a first location server, and the transfer of identifiers and location information to a second location server. RFC1034's disclosure of the storage and transfer of location information and identifiers, combined with Neimat's disclosure of the transfer of identifiers and associated information from a first server to a second server when a predetermined performance limit is reached, teach all limitations of [17.e]. (*Id.*, ¶207.)

⁹ One of ordinary skill would have also understood Neimat to teach other "predetermined performance limits" other than a memory location literally becoming full. As one of ordinary skill would have understood, computer engineers never allow a storage device to become *completely* full; some empty storage space is always necessary for the storage device to function properly. Thus, one of ordinary skill would have understood Neimat to teach predetermined performance limits of less than 100% capacity. (Ex. 1002, ¶205.) At a minimum, such predetermined performance limits would have been obvious in view of Neimat. (*Id.*)

Claim 23 depends from claim 17 and adds the following additional limitation: “*wherein the performance criterion comprises an amount of available persistent storage space in the first location server.*” Neimat discloses “persistent storage”: “Preferably, memory locations are in RAM, although they could alternatively be on disk as well.” (Ex. 1004, 8:42-44; Ex. 1002, ¶208.) One of ordinary skill would have understood “disk” to be a reference to persistent storage. (Ex. 1002, ¶208.) Neimat also discloses the performance criterion comprising “an amount of available storage space in the first location server.” As explained above, Neimat discloses the “performance criterion” being reached when a memory location becomes full. (*Supra*, at [17.e].)

Claim 30 depends from claim 17 and provides the further limitation of “*wherein transferring a portion of the identifiers and associated locations to a second location server when a performance criterion of the first data location server reaches a predetermined performance limit further comprises monitoring the performance criterion and automatically transferring the portion of identifiers and associated locations when the first location server reaches the predetermined limit.*”

Neimat discloses “monitoring the performance criterion.” There would be no possible way to conduct the “splitting” described in Neimat unless the performance criterion was “monitored,” such that the splitting would occur when the per-

formance criterion was met. (Ex. 1002, ¶211.) Moreover, Neimat discloses a “split coordinator server” that “receives a collision message over the network from a server that includes a memory location that undergoes a collision.” (Ex. 1004, 14:42-44.) Thus, the “split coordinator server” monitors whether a collision occurs as a result of the memory becoming full—*i.e.*, it monitors the “performance criterion.” (Ex. 1002, ¶212.)

Neimat also discloses “automatically transferring” the records when the first server “reaches the predetermined limit.” The “split coordinator server transmits a split message to a target server that transmits the predetermined memory location identified by the coordinator pointer value.” (Ex. 1004, 14:44-47.) “In response to the split message,” the “target server performs” several steps that result in splitting. (*Id.*, 14:51-15:15.) Thus, the transfer takes place automatically, *i.e.*, the server is programmed to automatically conduct this operation. (Ex. 1002, ¶213.) This transfer takes place when the memory location becomes full—*i.e.*, the predetermined limit is reached. (*Id.*, ¶¶212-213.)

Moreover, one of ordinary skill would not have designed a system in which a user manually monitors for collisions and manually conducts splitting in response to such collisions. Such a system would be inefficient and defeat the purpose of building a distributed system intended to improve performance. (*Id.*, ¶214.)

Thus, RFC1034 and Neimat, in combination, disclose every element of the additional limitation in claim 30. (*Id.*, ¶215.) RFC1034 discloses “transferring a portion of the identifiers and associated locations to a second location server.” Neimat discloses monitoring a performance criterion, and transferring identifiers and associated information automatically when that performance criterion reaches a predetermined limit.

2. One of Ordinary Skill Would Have Had Reason to Combine RFC1034 with Neimat.

One of ordinary skill would have been motivated to combine the RFC1034 DNS name server system with Neimat’s memory splitting technique.

RFC1034 describes how DNS is fundamentally concerned about performance and scaling, the same problems addressed by Neimat. In particular, RFC1034 states that DNS uses distributed name servers and local caches to increase the amount of memory and improve performance: “The sheer size of the database and frequency of updates suggest that it must be maintained in a distributed manner, with local caching to improve performance.” (Ex. 1003, 2.) RFC1034 discloses that this caching occurs to reduce name server load: “A very important goal of the resolver is to eliminate network delay and name server load from most requests by answering them from its cache of prior results.” (*Id.*, 29.)

Similarly, RFC1034 discloses the addition of name servers at different points in the network in order to increase network bandwidth, for example, to

avoid “using limited transatlantic bandwidth.” (*Id.*, 21.) Additionally, RFC1034 is, by design, “intentionally extensible.” (*Id.*, 1.) And, as noted above, RFC1034 discloses the transfer of RRs among location servers: “Whole zones can be transferred between name servers by transferring the RRs.” (*Id.*, 20.) Because RFC1034 already discloses the transfer of RRs from one location server to another, is expressly concerned about name server load, and is intentionally extensible, it would be natural to consider additional complementary mechanisms such as proposed in Neimat. (Ex. 1002, ¶221.)

Neimat discloses scaling a distributed network: “The present invention accommodates any number of clients and servers, and allows the database storage to extend to occupy any number of servers.” (Ex. 1004, 6:37-39.) The goal of that scaling is to improve performance: “[I]t is important for speed of operation that the data processor is able to locate and retrieve a particular data record as accurately and quickly as possible.” (*Id.*, 1:36-40.) In scaling the network, Neimat seeks to ensure efficient loading of multiple servers: “[A]ddressable storage space available to the database expands to occupy new servers linearly, one server at a time, and only when servers already used are efficiently loaded.” (*Id.*, 8:12-15.)

Neimat discloses efficient loading of multiple servers via load *balancing*, *i.e.*, its “memory splitting” technique that balances the load across multiple servers. “Address space is gradually increased in response to each collision by a process of

‘splitting’ a respective predetermined one of memory locations or ‘buckets.’” (*Id.*, 3:19-21.) Splitting allows the system to scale “so that records reside on virtually any number of servers” and “to a size limited only by data storage capacity of the servers.” (*Id.*, 5:13-19.) “In accordance with such linear expansion principles, the table continues expanding as long as required.” (*Id.*, 3:44-46.)

In sum, RFC1034 already discloses the transfer of RRs from one location server to another, is expressly concerned about server load and performance, and is extensible. Neimat discloses one way of transferring records from one server to another in order to ensure servers are efficiently loaded: transferring when a predetermined storage criterion is reached. One of ordinary skill would have understood Neimat to be a solution to the problem of server load disclosed in RFC1034 and would combine these two references for at least this reason. (Ex. 1002, ¶223.)

One of ordinary skill would have also combined Neimat with RFC1034 for another reason: Neimat proposes a solution to the well-known problem of memory “collisions,” which is a problem experienced in distributed data storage systems such as the system disclosed in RFC1034. Neimat discloses that its memory splitting technique solves the “well known” problem in the art of a “collision.” (Ex. 1004, 13:27-28; Ex. 1002, ¶224.) This problem occurs, according to Neimat, when a memory location becomes filled to a limit and can no longer store additional data. (See Ex. 1004, 13:26-42, 2:38-44.) There is a “collision” problem because the

system wants to put data in a memory location, but the location is already occupied. (*See id.*) Neimat discloses that “splitting” was a known solution to this known problem. (*See id.*, 13:32-37.)

One of skill would have appreciated that this problem and the splitting solution in Neimat would have had direct application in DNS as described by RFC1034. DNS would have a similar “collision” problem in the process of adding domain names. (*See Ex. 1002, ¶224.*) As in Neimat, if a DNS name server became full and the DNS system attempted to put data in that server, then a “collision” would occur. (*Id.*) That first name server could no longer store data and the DNS system must determine a second name server to store the data. (*Id.*) Presented with this known problem reflected by RFC1034, one of skill would have been motivated to use the Neimat splitting solution to solve that same problem. (*Id.*)

The combination of RFC1034 and Neimat would have been a straightforward combination of known prior art elements with predictable results as well as a reasonable expectation of success. *See KSR*, 550 U.S. at 416 (“The combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results.”); M.P.E.P. § 2143(I)(A). Dr. Weissman explains how one of skill easily would have been able to modify DNS, as disclosed in RFC1034, to incorporate the splitting technique of Neimat. (*Ex. 1002, ¶¶225-*

231.) Programming a DNS server with Neimat's splitting technique would have been a basic task for programmer system administrator, for instance. (*Id.*, ¶230)

D. Ground 4: The Combination of RFC1034, Neimat, and Venkatasubramanian Renders Obvious Claim 24.

RFC1034 (Ex. 1003) in view of Neimat (Ex. 1004) further in view of Venkatasubramanian (Ex. 1005) renders obvious claim 24.

Claim 17, from which Claim 24 depends, recites that load balancing occurs when a “*predetermined performance limit*” is reached. Claim 24 provides the further limitation of “*the performance criterion comprises a transaction rate limit.*”

Venkatasubramanian describes a “Load Management Example” with different scenarios in which a server manages load based on a performance criterion reaching a transaction rate limit. (*See* Ex. 1005, 2.) That “Load Management Example” begins as follows:

Suppose that at some time instant, a video server that has two data sources S_1 and S_2 has sufficient storage to accommodate eight video objects on S_1 and two video objects on S_2 . Further, suppose S_1 and S_2 have sufficient bandwidth to support three and eight additional requests, respectively. Also, suppose that ten additional requests, eight for video object V_1 and one for each video objects V_2 and V_3 are expected to arrive in the near future. Considering the disk space and bandwidth availabilities of the data sources, it is prudent to allocate V_1 to S_2 and V_2 and V_3 to S_1 . On the other hand, if V_2 and V_3 were allocated to S_2 , the video server would only be able to handle five of the ten requests – three for V_1 and one each for V_2 and V_3 [I]f additional requests are received for V_1 , replication becomes essential. In general, the exact load

management mechanism has to be decided depending upon the current utilization of resources in the data sources of the video server and the expected subscriber requests including buffer memory and CPU processing power.

(*Id.*, 2 (emphasis added).)

In this example, the phrase “bandwidth to support three and eight additional requests” refers the rate at which requests can be serviced. (Ex. 1002, ¶236.) Because each “request” corresponds to a transaction, this is a “transaction rate limit.” (*Id.*)

The claims and specification of the '978 patent support the proposition that the statement “S₁ and S₂ have sufficient bandwidth to support three and eight additional requests, respectively” discloses a transaction rate limit. Dependent claim 26 refers to a “network connection bandwidth limit” as a type of “transaction rate limit.” Similarly, the '978 patent specification explains that “an NDTP server cluster 50 linearly increases... transaction rate[] by adding processing power and network bandwidth.” (Ex. 1001, 15:58-63.)

As explained above, Neimat and RFC1034, together, disclose all elements of claim 17, including “transferring a portion of the identifiers and associated locations to a second data location server when a performance criterion of the first location server reaches a predetermined performance limit.” Venkatasubramanian discloses a type of performance criterion that is used in load balancing of distribut-

ed systems: a transaction rate limit. Thus, Venkatasubramanian discloses the dependent limitation of claim 24: “the performance criterion comprises a transaction rate limit.” For this reason, Venkatasubramanian, in combination with Neimat and RFC1034, disclose all elements of claim 24.

Moreover, in addition to disclosing a “performance criterion” comprising a “transaction rate limit,” Venkatasubramanian also discloses the transferring of data from one server to a different server when that transaction rate limit is reached. Venkatasubramanian recites: “[i]f additional requests are received for V_I , replication becomes essential.” (Ex. 1005, 2.) In other words, when the transaction rate limit for the server storing V_I is reached, it becomes essential to engage in “replication,” i.e., to transfer data from one server to a different server. (Ex. 1002, ¶237.) Therefore, Venkatasubramanian discloses that when a “transaction rate limit” is reached, data is transferred to a different server for purposes of load balancing. (*Id.*)

It would have been obvious to one of ordinary skill to combine Venkatasubramanian with Neimat and RFC1034. Neimat’s “splitting” is one load management technique based on whether a storage location becomes filled to a limit. (*See, e.g.*, Ex. 1004, 13:26-42.) As previously explained, one of ordinary skill would have combined Neimat’s “splitting” disclosure with RFC1034.

Venkatasubramanian likewise teaches load management based on the amount of “disk space.” (*See, e.g.*, Ex. 1005, 2.) But Venkatasubramanian also teaches that load management based on storage space (as in Neimat) is only one approach, and that a combination of that approach with others is preferred. Specifically, Venkatasubramanian states that “disk bandwidth, memory buffer space, CPU cycles, and network transfer bandwidth” are different variables that may be “necessary for supporting” a request. (*Id.*, 529, 2.) Venkatasubramanian explicitly teaches the combination of these different approaches for system optimization: “Performance evaluations indicate that a load management procedure which uses a judicious combination of the different policies performs best for most server configurations.” (*Id.*, 1.) Thus, Neimat teaches one method of load balancing (based on disk space), and Venkatasubramanian expressly teaches Neimat’s method while presenting methods of load balancing based on other performance criteria, including transaction rate, as alternatives to Neimat’s method.

One of ordinary skill would have therefore modified Neimat in view of Venkatasubramanian’s teaching. (Ex. 1002, ¶241.) And, one of ordinary skill would have been motivated to combine RFC1034 and Neimat as described above. Dr. Weissman explains how one of skill easily would have been able to modify Neimat, as disclosed in Venkatasubramanian, to incorporate additional load balancing techniques. (*Id.*, ¶242) Therefore, there is ample reason to combine

RFC1034, Neimat, and Venkatasubramanian, with a reasonable expectation of success.

VII. LACK OF SECONDARY CONSIDERATIONS EVIDENCE

Petitioner is not aware of any evidence of secondary considerations with a nexus to the claimed invention that supports patentability. Even if there were secondary considerations, such evidence “simply cannot overcome a strong prima facie case of obviousness,” as is present here. *Wyers v. Master Lock Co.*, 616 F.3d 1231, 1246 (Fed. Cir. 2010) (“[W]here the inventions represen[t] no more than ‘the predictable use of prior art elements according to their established functions,’... the secondary considerations are inadequate to establish nonobviousness as a matter of law.”). If Patent Owner raises arguments alleging secondary considerations, Petitioner requests an opportunity to respond.

VIII. MANDATORY NOTICES

A. Real Party-in-Interest [37 C.F.R. §42.8(b)(1)]

Petitioner Amazon Web Services, Inc. is a real party-in-interest for this proceeding. In addition, for completeness, Petitioner identifies Amazon.com, Inc., Amazon.com Services, Inc., Amazon Digital Services, LLC, and AWSHC, Inc. as real parties-in-interest for this proceeding. No other parties have controlled, directed, funded, or participated in the preparation of this Petition, and therefore no other parties are understood to be a real party-in-interest here.

B. Related Matters [37 C.F.R. §42.8(b)(2)]

Patent Owner asserted the '978 patent against Petitioner in the Northern District of Illinois on December 12, 2018 (No. 1:18-cv-08175), served on December 14, 2018. Patent Owner asserts in that case the same claims at issue here.

C. Lead/Back-up Counsel [37 C.F.R. §42.8(b)(3)]

Petitioner is filing a power of attorney designating Michael G. Babbitt (Reg. No. 59,288, mbabbitt@jenner.com) lead counsel and Lisa M. Schoedel (Reg. No. 53,564, lschoedel@jenner.com) back-up counsel, both of Jenner & Block LLP, 353 N. Clark. St., Chicago, Illinois 60654.

D. Service Information [37 C.F.R. §42.8(b)(4)]

Petitioner consents to e-mail service at the addresses of lead and back-up counsel and chgoip@jenner.com. Hand delivery is also available to the addresses of lead and back-up counsel.

IX. GROUNDS FOR STANDING [37 C.F.R. §42.104(A)]

Petitioner certifies that the '978 patent is available for *inter partes* review, and Petitioner is not barred or estopped from challenging the claims on the grounds identified herein.

X. CONCLUSION

For the foregoing reasons, Petitioner respectfully requests that the Board institute *inter partes* review trial and that claims 1, 3, 6, 10, 14, 17, 23, 24, 30, and 31 of the '978 patent be canceled as unpatentable.

Respectfully submitted,

December 12, 2019

/s/ Michael G. Babbitt
Counsel for Petitioner

CERTIFICATE OF COMPLIANCE

The undersigned certifies that this brief complies with the applicable type-volume limitations of 37 C.F.R. § 42.24. Exclusive of the portions exempted by 37 C.F.R. § 42.24(a), this brief contains 13,120 words as counted by the word processing program used for its preparation (Microsoft Word 2013).

/s/ Michael G. Babbitt

Counsel for Petitioner

CERTIFICATE OF SERVICE

I hereby certify that on December 12, 2019, I caused a true and correct copy of the foregoing materials:

- PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 7,233,978;
- PETITIONER’S EXHIBITS 1001-1041; AND
- PETITIONER’S POWER OF ATTORNEY.

to be served via USPS Priority Mail Express to the Patent Owner’s Attorney of

Record PAIR Correspondence Address for U.S. Patent No. 7,233,978 as follows:

Kent Genin (Reg. No. 37834)
Brinks Hofer Gilson & Lione
P.O. Box 10395
Chicago, IL 60610

As well as via electronic mail to counsel of record in the related N.D. Ill. district court action No. 1:18-cv-08175.

/s/ Michael G. Babbitt
Counsel for Petitioner