UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner

v.

KOVE IO, INC.,
Patent Owner

Patent No. 7,103,640

PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 7,103,640

TABLE OF CONTENTS

I. INTRODUCTION 1
II. MANDATORY NOTICES 1
III. PAYMENT OF FEES UNDER 37 C.F.R. §§ 42.15 AND 42.103 3
IV. GROUNDS FOR STANDING 3
V. PRECISE RELIEF REQUESTED AND GROUNDS RAISED 3
VI. LEVEL OF ORDINARY SKILL 5
VII. OVERVIEW OF THE '640 PATENT 5
VIII. CLAIM CONSTRUCTION 8
IX. DETAILED EXPLANATION OF GROUNDS 11
A. Ground 1: Kahn Anticipates Claims 10-13, 17-19, and 24 11
1. Claim 10 11
2. Claim 11 30
3. Claim 12 31
4. Claim 13 32
5. Claim 17 33
6. Claim 18 37
7. Claim 19 43
8. Claim 24 44
B. Ground 2: Kahn in View of Vingralek Renders Claims 10-24 Obvious 45
1. Claim 10 45
2. Claim 11 54
3. Claim 12 55
4. Claim 13 56
5. Claim 14 56
6. Claim 15 56
7. Claim 16 57

8. Claim 17 58
9. Claim 18 59
10. Claim 19 60
11. Claim 20 61
12. Claim 21 61
13. Claim 22 61
14. Claim 23 62
15. Claim 24 62

X. DISCRETIONARY DENIAL IS NOT APPROPRIATE 62
A. The Board Should Not Deny Institution Under § 325 62
B. Discretionary Denial Under *General Plastic* Is Not Appropriate 65
C. The Board Should Not Deny Institution Under *Fintiv* 66

XI. CONCLUSION 69

XII. APPENDIX A (CLAIM LISTING) 70

LIST OF EXHIBITS

Ex. 1001	U.S. Patent No. 7,103,640 to Overton et al. ("the '640 patent")
Ex. 1002	Declaration of Sandeep Chatterjee, Ph.D.
Ex. 1003	Curriculum Vitae of Sandeep Chatterjee, Ph.D.
Ex. 1004	Prosecution History of U.S. Patent No. 7,103,640
Ex. 1005	WIPO Application No. 1997/043717 to Kahn et al.
Ex. 1006	U.S. Patent No. 6,473,781 to Skagerwall et al.
Ex. 1007	R. Vingralek, *Distributed file organization with scalable cost/performance* (1993)
Exs. 1008-1010	RESERVED
Ex. 1011	Japanese Patent Application Publication No. H09-231180 to Sato et al.
Exs. 1012-1013	RESERVED
Ex. 1014	RFC 768, User Datagram Protocol (Aug. 28, 1980)
Ex. 1015	M. C. Norrie et al., *The Database Research Group at ETH Zurich*, SIGMOD Record (March 1994)
Ex. 1016	U.S. Patent No. 5,764,756
Exs. 1017-1018	RESERVED
Ex. 1019	First Amended Complaint (Dkt. No. 36) in *Kove IO, Inc. v. Google LLC*, Case No. 1:23-cv-04244 (N.D. Ill. Nov. 14, 2023)
Ex. 1020	Declaration of Sylvia D. Hall-Ellis, Ph.D.
Ex. 1021	RESERVED

Ex. 1022	*Amazon Web Services, Inc. v. Kove IO, Inc.*, IPR2020-00276, Paper 23 (June 16, 2020)
Ex. 1023	RESERVED
Ex. 1024	Memorandum Opinion and Order (Dkt. No. 484) in *Kove IO, Inc. v. Amazon Web Services, Inc.*, Case No. 1:18-cv-08175 (N.D. Ill. Dec. 17, 2021)
Ex. 1025	Memorandum Opinion and Order (Dkt. No. 739) in *Kove IO, Inc. v. Amazon Web Services, Inc.*, Case No. 1:18-cv-08175 (N.D. Ill. Feb. 6, 2024)
Ex. 1026	Request for *Ex Parte* Reexamination of U.S. Patent No. 7,103,640 in *Ex Parte* Reexamination No. 90/014,552 (July 23, 2020)
Ex. 1027	Order Denying Request for *Ex Parte* Reexamination in *Ex Parte* Reexamination No. 90/014,552 (Sept. 9, 2020)
Ex. 1028	Request for *Ex Parte* Reexamination of U.S. Patent No. 7,103,640 in *Ex Parte* Reexamination No. 90/019,036 (Nov. 19, 2021)
Ex. 1029	Order Granting Request for *Ex Parte* Reexamination in *Ex Parte* Reexamination No. 90/019,036 (Dec. 23, 2021)
Ex. 1030	Notice of Intent to Issue *Ex Parte* Reexamination Certificate in *Ex Parte* Reexamination No. 90/019,036 (Sept. 22, 2022)
Ex. 1031	Request for *Ex Parte* Reexamination of U.S. Patent No. 7,103,640 in *Ex Parte* Reexamination No. 90/019,166 (Feb. 17, 2023)
Ex. 1032	Order Granting Request for *Ex Parte* Reexamination in *Ex Parte* Reexamination No. 90/019,166 (Apr. 4, 2023)
Ex. 1033	2020 to 2023 Northern District of Illinois Orders Regarding Motions to Stay
Ex. 1034	Agreed Order Setting Amended Patent Case Schedule (Dkt. No. 49) in *Kove IO, Inc. v. Google LLC*, Case No. 1:23-cv-04244 (N.D. Ill. Feb. 20, 2024)

Ex. 1035	Federal Court Management Statistics (December 2023)
Ex. 1036	Memorandum by Director Vidal (June 21, 2022)
Exs. 1037-1051	RESERVED
Ex. 1052	Order for Staying Discovery Process until July 15, 2024 (Dkt. No. 65) in *Kove IO, Inc. v. Google LLC*, Case No. 1:23-cv-04244 (N.D. Ill. May 1, 2024)
Ex. 1053	Northern District of Illinois Local Patent Rules
Ex. 1054	RESERVED
Ex. 1055	Google's Motion to Stay and for Expedited Briefing Schedule (Dkt. No. 64) in *Kove IO, Inc. v. Google LLC*, Case No. 1:23-cv-04244 (N.D. Ill. Apr. 30, 2024)
Ex. 1056	Joseph B. Greene, *Oracle8 Server Unleashed* (1998)
Ex. 1057	Joseph B. Greene, *Oracle DBA Survival Guide* (1995)
Ex. 1058	Betty Salzberg, *File Structures* (1988)
Ex. 1059	Andrew P. Black & Yeshayahu Artsy, *Implementing Location Independent Invocation* (1989)

I. INTRODUCTION

Google LLC ("Petitioner") respectfully requests *inter partes* review of claims 10-24 ("the challenged claims") of U.S. Patent No. 7,103,640 ("the '640 patent") (Ex. 1001) assigned to Kove IO, Inc. ("Patent Owner").

II. MANDATORY NOTICES

Real Parties-in-Interest: Pursuant to 37 C.F.R. § 42.8(b)(1), Petitioner identifies Google LLC as the real party-in-interest.[1]

Related Matters: The '640 patent is asserted in *Kove IO, Inc. v. Amazon Web Services, Inc.*, Case No. 1:18-cv-08175 (N.D. Ill.) ("the AWS Proceeding"), and *Kove IO, Inc. v. Google LLC*, Case No. 1:23-cv-04244 (N.D. Ill.). The '640 patent was involved in two ex-parte reexaminations, Nos. 90/014,552 ("the '552 EPR") and 90/019,036 ("the '036 EPR"), and is currently pending in ex-parte reexamination No. 90/019,166 ("the '166 EPR").

The '552 EPR was filed by a third party (Ex. 1026) in July 2020, presenting SNQs on claims 17, 18, and 24 based on different combinations of prior art from the grounds here (*id.*, 21-24; *infra* Sections IV, IX); the request was denied in

[1] Google LLC is a subsidiary of XXVI Holdings Inc., which is a subsidiary of Alphabet Inc. XXVI Holdings Inc. and Alphabet Inc. are not real parties-in-interest to this proceeding.

September 2020 (Ex. 1027, 1, 38). The '036 EPR was filed by third party Amazon Web Services ("AWS") (Ex. 1028) in November 2021, presenting SNQs on claims 17, 18, and 24 based on different combinations of prior art from the grounds here (*id.*, 11); the request was granted (Ex. 1029, 21) and the claims were confirmed (Ex. 1030, 6). Petitioner had no involvement in the preparation or filing of either EPR.

The '166 EPR was filed by third party AWS (Ex. 1031) in February 2023, presenting SNQs on claims 17, 18, and 24 based on different combinations of prior art from the grounds here (*id.*, 19; *infra* Sections IV, IX.) Petitioner had no involvement in the preparation or filing of the '166 EPR. Because the request raised substantial new questions of patentability, reexamination was ordered in April 2023. (Ex. 1032, 20.) The examiner has not issued an Office Action at the time of the filing of the current Petition.

Petitioner is concurrently filing IPR petitions challenging related U.S. Patent Nos. 7,814,170 ("the '170 patent") and 7,233,978 ("the '978 patent"), which are also at issue in the above-noted district court cases. The '978 patent was also at issue in *Amazon Web Services, Inc. v. Kove IO, Inc.*, IPR2020-00276 ("the AWS IPR").

Counsel and Service Information: Lead counsel is Naveen Modi (Reg. No. 46,224). Backup counsel are Joseph E. Palys (Reg. No. 46,508), Daniel

Zeilberger (Reg. No. 65,349), Howard Herr (*pro hac vice* to be filed), Alexa Lowman (*pro hac vice* to be filed), and Puja Patel (*pro hac vice* to be filed). Service information is Paul Hastings LLP, 2050 M Street NW, Washington, D.C. 20036, Tel.: 202.551.1700, Fax: 202.551.1705, email: PH-Google-Kove-IPR@paulhastings.com. Petitioner consents to electronic service.

III. PAYMENT OF FEES UNDER 37 C.F.R. §§ 42.15 AND 42.103

The PTO is authorized to charge any fees due during this proceeding to Deposit Account No. 50-2613.

IV. GROUNDS FOR STANDING

Petitioner certifies under 37 C.F.R. § 42.104(a) that the '640 patent is available for review and Petitioner is not barred or estopped from requesting review on the grounds identified herein.

V. PRECISE RELIEF REQUESTED AND GROUNDS RAISED

Petitioner requests review of claims 10-24 of the '640 patent and cancellation of these claims as unpatentable on the following grounds:

Ground 1: Claims 10-13, 17-19, and 24 are unpatentable under pre-AIA 35 U.S.C. § 102 as being anticipated by Kahn (Ex. 1005).

Ground 2: Claims 10-24 are unpatentable under pre-AIA 35 U.S.C. § 103 as being obvious over Kahn and Vingralek (Ex. 1007).

The earliest possible effective filing date for the '640 patent is July 8, 1998.[2]

Kahn published on November 20, 1997 from PCT application U.S. 1997/007834 filed on May 9, 1997. (Ex. 1005, Cover). Thus, Kahn qualifies as prior art at least under pre-AIA 35 U.S.C. § 102(a).

Vingralek is a journal article published by ETH Zürich as early as December 1993, and thus is prior art under at least pre-AIA 35 U.S.C. § 102(b). The declaration of Dr. Sylvia D. Hall-Ellis (Ex. 1020, ¶¶1-66) confirms that Vingralek was published, disseminated, and reasonably available to the public on or shortly after December 31, 1993. (Ex. 1020, ¶¶46-54; *see also id.*, ¶¶19-45.) For example, Dr. Hall-Ellis explains that based on finding a copy of Vingralek that is dated December 1993 in the ETH Zürich Library and a MARC record in its online library catalog indicating that the ETH Zurich journal was first published in 1974. (*Id.*, ¶¶51-54.) Moreover, Vingralek was cited by Ex. 1015 ("Norrie"), which was published on March 1, 1994. (*Id.*, ¶¶55-63.) It is her opinion that Vingralek was

[2] Petitioner believes the '640 patent is not entitled to a priority date earlier than September 14, 1999, but does not raise that issue in this petition as it is not relevant. Petitioner does not concede that the '640 patent is entitled to either the July 8, 1998 or September 14, 1999 priority date.

publicly available on or shortly after December 31, 1993, as shown on the cover. (*Id.*, ¶¶64-65.)

VI. LEVEL OF ORDINARY SKILL

A person of ordinary skill in the art at the time of the alleged invention of the '640 patent ("POSITA") would have had an undergraduate degree in electrical engineering, computer engineering, computer science or a related field along with two years of work experience in the field of distributed computing systems. (Ex. 1002, ¶¶20-21.)[3] More education can supplement practical experience and vice versa. (*Id*.)

VII. OVERVIEW OF THE '640 PATENT

The '640 patent describes storing and retrieving information across a network (Ex. 1001, 1:26-30, 1:56-58) and a network distributed tracking protocol (NDTP) (*id.*, 2:12-14, 4:25-36). Figure 11 shows an NDTP server constellation where a client consults servers that return strings in response to a client request. (*Id.*, 16:53-56.)

[3] Petitioner submits the declaration of Dr. Sandeep Chatterjee (Ex. 1002), an expert in the field of the '640 patent. (Ex. 1002, ¶¶1-28; Ex. 1003.)

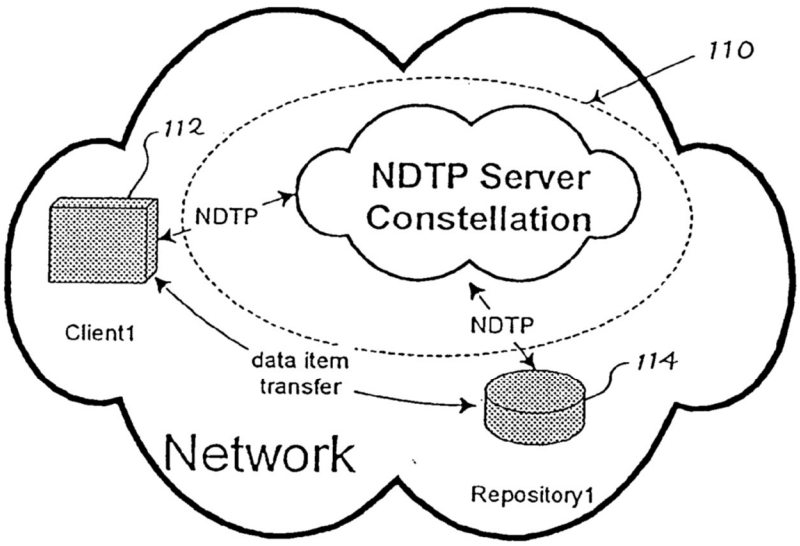
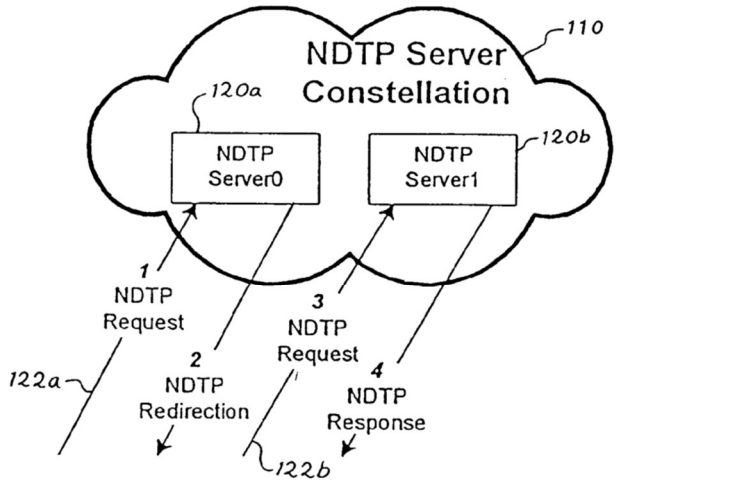


Fig. 11 NDTP Server Constellation Context

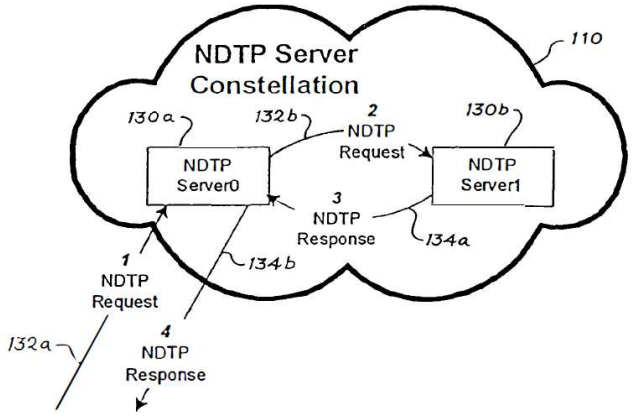
(*Id.*, FIG. 11; Ex. 1002, ¶29.)

Server constellations have two organizational approaches: client-centric (Figure 12) and server-centric (Figure 13). In the client-centric approach, processing emphasis is toward clients (*id.*, 18:11-13), whereas in the server-centric approach, server processing is emphasized (*id.*, 18:37-41).



NDTP Client-Centric Constellation

Fig. 12



NDTP Server-Centric Constellation

Fig. 13

(*Id.*, FIGS. 12, 13; Ex. 1002, ¶30.)

In both approaches, a client transmits an identifier specifying an entity to a server in a client request. (Ex. 1001, 2:12-44.) If the server contains location information specifying network locations of data associated with the entity, the server returns the location information. (*Id.*; Ex. 1002, ¶31.)

As demonstrated below, all of the features recited in the challenged claims of the '640 patent were known in the prior art before the alleged invention date of the '640 patent. (*Infra* Section IX; Ex. 1002, ¶¶32-38, 22-28; Exs. 1006, 1011, 1056-1059.)

VIII. CLAIM CONSTRUCTION

The Board only construes the claims when necessary to resolve the underlying controversy. *Toyota Motor Corp. v. Cellport Sys., Inc.*, IPR2015-00633, Paper No. 11 at 16 (Aug. 14, 2015). For purposes of this proceeding, Petitioner believes that no special constructions are necessary to assess whether the challenged claims are unpatentable over the asserted prior art. (Ex. 1002, ¶39.)[4]

As shown below, in the AWS IPR, the Board construed a claim term of the related '978 patent (Ex. 1022, 13), and in the AWS Proceeding, the district court construed several claim terms of the '640 patent. (Ex. 1024, 35; Ex. 1025, 19.)

[4] Petitioner reserves all rights to raise claim construction and other arguments in other venues.

While Petitioner does not acquiesce to those constructions, the prior art mappings in Section IX explain how the '640 patent claims are disclosed and/or rendered obvious if those constructions are applied. (Ex. 1002, ¶¶39-45.)

'640 Patent Terms	AWS Proceeding/IPR Construction
"identifier"[5]	"a unique encoding that identifies an individual entity, and with which zero or more data location specifiers are associated in a location server" (Ex. 1022, 13.)
"location information"	"information pertaining to one or more locations of data and/or the identities of one or more location servers" (Ex. 1024, 35.)
"location server"	"a network-attached component that maintains a set of identifier/location mappings that are modified or

[5] Petitioner applies the construction of "identifier" recited in the '978 patent claims to '640 patent claims reciting "identification string"/"identifier string." because the '640 patent uses the three terms interchangeably (Ex. 1001, 2:19-24, 4:7-9) and describes the terms in substantially the same manner as the '978 patent. (*Compare* Ex. 1001, 4:7-9 *with* Ex. 1022, 10-11.)

'640 Patent Terms	**AWS Proceeding/IPR Construction**
	returned in response to location request messages from clients" (Ex. 1024, 35.)
"client"	"a network-attached component (which may be software or hardware) that initiates update or lookup of identifier/location mappings from a location server with location request messages" (Ex. 1024, 35.)
"plurality of location servers"[6]	"location servers in non-hierarchical structures" (Ex. 1025, 19.)

[6] Petitioner also applies this construction to claims reciting "plurality of data location servers."

IX. DETAILED EXPLANATION OF GROUNDS

As detailed below, each of the challenged claims are unpatentable.

A. Ground 1: Kahn Anticipates Claims 10-13, 17-19, and 24

1. Claim 10

i) [10.pre]

To the extent the preamble is limiting, Kahn discloses the limitations therein. (Ex. 1002, ¶¶47-52.) To begin, Kahn discloses a method for managing digital objects stored in one or more repositories in a network, where the digital objects are located by one or more pointers associated with a handle. (Ex. 1005, Abstract ("A method of managing digital objects…in a network is presented."), 1:10-12 ("By a 'digital object' we broadly mean any set of sequences of bits or digits and an associated unique identifier which we call a 'handle.'"), 6:9-10, 6:24.) Multiple handle servers in the network each accept handles of digital objects and return corresponding pointers indicating the digital objects' locations in the network. (*Id.*, Abstract, 6:28-35, 7:2-20, 21:6-24; Ex. 1002, ¶47.) For example, Kahn's Figure 1 shows an exemplary system that includes a repository (blue) for storing digital objects (green) in the network, where the digital objects are identified by their handle (yellow) and associated pointers. (Ex. 1005, Abstract.) Handle-pointer mappings are stored in tables within handle servers (red). (*Id.*, Abstract,

7:19-20 ("…each handle server having a table which associates handles with pointers."), 11:16-18, 37:31-33.)

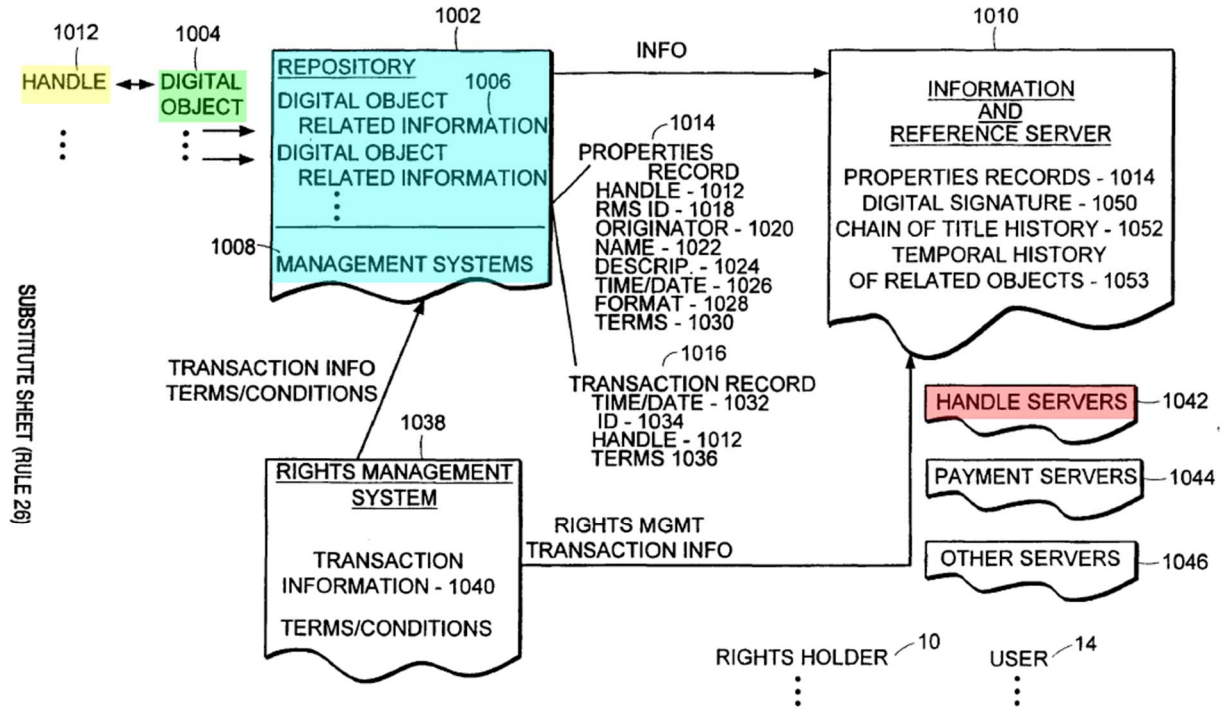


FIG. 1

(*Id.*, FIG. 1; Ex. 1002, ¶47.)

Kahn discloses that digital objects (including "data") are "stored in a distributed network." A digital object is represented by a sequence of bits or digits, may include conventional digital representations of works (e.g., books, images, sounds, software), and is identified by a unique handle. (Ex. 1005, 1:10-12, 1:16-20, 21:3-5 ("Each digital object has **a 'handle', a concise unique identifier for a digital object**….").) A POSITA would have understood that sequences of bits/digits associated with each digital object are "data." (Ex. 1002,

¶48.) Indeed, Kahn discloses that "a digital object could include programs and data" which enable delivery over a network or reproduction on a computer screen. (Ex. 1005, 1:20-25.) Kahn discloses that a digital object is stored in a repository in a distributed network, such as the Internet, which does not have a central nodal point. (*Id.*, Abstract ("**The [digital] objects are stored at locations…accessible in the network.**"), 2:25-30 ("digital objects available on networks (e.g., **Internet**)"), 6:9-35, 11:16-18, 21:21-24 ("services can be **provided on a widely distributed basis**"), 73:23-31 ("One or more pointers, once returned, identifies the network location of the one or more repositories….[The] **network [is] without a central nodal point** in the system…."); Ex. 1001, 1:56-65, 4:49-57; Ex. 1002, ¶48.)

Kahn explains that one or more pointers ("data location information") returned by a handle server indicate a stored digital object's location in the distributed network. (Ex. 1005, Abstract ("[M]ultiple servers…accept[] identifiers of…digital objects and return[] corresponding **pointers to the locations of the digital objects in the network**…."), 3:21-22, 6:30-35, 28:32-34.) Kahn also discloses the claimed "data location information" under the AWS Proceeding's construction of "location information" because the pointers provide information pertaining to one or more locations of digital objects (including "data"). (Section VIII; Ex. 1002, ¶49.)

Kahn discloses "retrieving data location information for data" stored in the distributed network. Figure 8 (below) shows a client system retrieving one or more pointers ("data location information") for digital objects (including "data"). (Ex. 1005, FIG. 8, 46:26-28, 46:29-47:25, 72:32-74:32.)

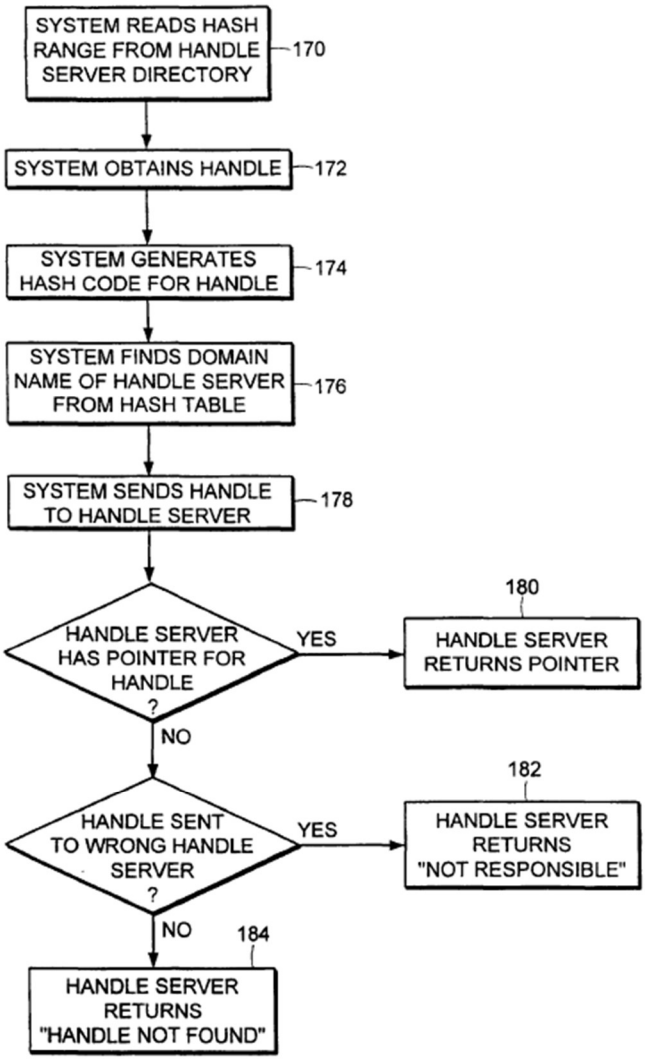


FIG. 8
SUBSTITUTE SHEET (RULE 26)

(*Id.*, FIG. 8.) In step 170, a client downloads from a handle directory server a hash range/handle-server-domain-name table (also referred to as a hash code/handle

server table) that associates hash ranges with handle server domain names. (*Id.*, 46:22-47:5, FIG. 7.) The table allows the client to determine the location of a handle server that contains a pointer for retrieving the digital object associated with the handle. (*Id.*, 29:18-26, 38:9-17, 46:22-47:5 ("The system…consults the hash range/handle-server-domain-name table to determine the domain name of the appropriate handle server (step 176)."), 73:4-74:26.)

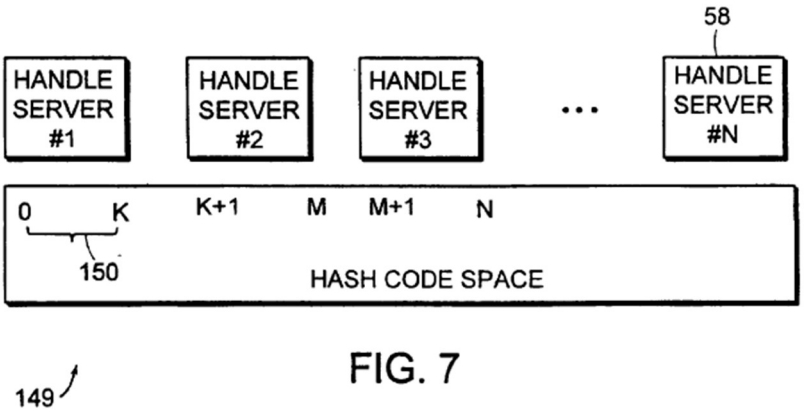


FIG. 7

(*Id.*, FIG. 7; Ex. 1002, ¶50.) For instance, as shown in Figure 7, each handle server is assigned a range of hash codes/values. (Ex. 1005, 46:22-24, FIG. 7.) The client uses a generated hash code/value (discussed below) as an index into the downloaded table, allowing the client to obtain the domain name of the handle server that stores the pointers for locating the digital object associated with the handle. (*Id.*, 47:1-11, 73:17-26; Ex. 1002, ¶50.)

Returning to Figure 8, after downloading the hash range/handle-server-domain-name table in step 170, the client in step 172 receives a handle from a

user's request (e.g., via bibliographic search or barcode scan) for obtaining a digital object. (Ex. 1005, 46:34-47:1, 73:8-16.) Subsequently, the client "generates the hash code for the handle using a predetermined hashing algorithm," then uses the hash code to index into the hash range/handle-server-domain-name table "to determine the domain name of the appropriate handle server," and sends the handle to that server in steps 174, 176, and 178. (*Id.*, 47:1-11, 73:17-26; Ex. 1002, ¶51.)

Next, the handle server verifies if it does or, e.g., in the event of a dynamic change to the downloaded table, does not contain the handle and its associated pointers. (Ex. 1005, 47:10-11, 73:32-74:12 ("The handle server verifies that the handle falls within the set of handles for whose hash value it is responsible… [or] [i]f it is not in this set, due to some dynamic system change or error condition….").) If the handle is indeed located in the queried handle server, one or more pointers for locating the digital object stored in the distributed network are returned to the client in step 180 ("retrieving data location information for data stored in a distributed network"). (*Id.*, 29:8-10, 47:10-13 ("If the handle server found the pointers, **a list of pointers is returned (step 180)**."), 73:23-26, 74:21-26.) The client then uses the retrieved pointers to obtain the digital object stored in a repository in the distributed network. (*Id.*, 74:21-29 ("For each pointer returned by the handle server, the requesting system uses the pointer to attempt to obtain a

copy of the digital object….").) Accordingly, Kahn discloses this limitation. (Ex. 1002, ¶52.)

ii) [10.a]

Kahn discloses this limitation. (Ex. 1002, ¶¶53-61.) **First**, Kahn discloses "data associated with an identification string." Kahn discloses that each digital object (including "data") is associated with a unique handle ("identification string") identifying that digital object, where the handle comprises a "unique character string[]." (Ex. 1005, 7:4-5; *see also id.*, 6:27-28, 9:23-26, 11:8-13, 21:3-5; Section IX.A.1.i; Ex. 1001, 2:16-18, 2:49-51, 2:52-53; Ex. 1002, ¶53.) Kahn's handle is also an "identification string" under the AWS IPR construction of related term "identifier" because the unique character string (i.e., unique encoding) identifies a digital object (i.e., individual entity). (Section VIII.) A POSITA would have understood that the handle is a unique encoding because each digital object is uniquely represented and encoded in the system by the characters that comprise the handle. (Ex. 1002, ¶54; Ex. 1005, 33:3-14, 33:22-27.) Kahn also explains that a handle may be associated with zero or more pointers (i.e., zero or more data location specifiers) in a handle server (i.e., a location server, as construed in the AWS Proceeding and discussed below), and the pointer specifies the location of the digital object (including "data"). (Ex. 1005, 28:31-34; *see also id.*, 7:25-27, 37:31-33.)

Second, Kahn discloses "receiving at a first client a data query for retrieving data associated with an identification string." (Ex. 1002, ¶55.) As discussed for limitation [10.pre], the client ("first client") receives a request ("data query") from a user, using a digital object's handle as input to the client, for retrieving a digital object (including "data") associated with the handle ("identification string") in the request. (Ex. 1005, 73:8-74:32; Section IX.A.1.i.) Specifically, the client generates the hash code for the handle using a hashing algorithm, and uses the hash code to index into the hash range/handle-server-domain-name table to determine the domain name of the handle server that may contain a pointer to the digital object. (Ex. 1005, 47:1-5, 73:17-22; Section IX.A.1.i.) The client sends a request including the handle to the handle server, and if the handle server contains the handle and associated pointers, the pointers are returned to the client for it to locate and retrieve the associated digital object. (Ex. 1005, 47:5-11, 74:23-26.)

Kahn discloses the claimed "first client" under the AWS Proceeding's construction of "client." (Section VIII; Ex. 1001, 4:11-14.) Kahn discloses a user providing a handle to a client ("first client") to obtain a pointer indicating the digital object's location in the distributed network. (Ex. 1005, 46:25-47:25, 73:1-74:32.) Moreover, the client's user interface may be "in the form of software running on a workstation" (*id.*, 26:28-34, FIG. 2), and discloses that the user interface is part of a system that operates on a network (*id.*, 4:23-32, 19:26-31,

26:28-28:28, FIG. 2). Thus, a POSITA would have recognized that the user's computer/client ("first client") is a hardware and/or software component attached to a network. (Ex. 1002, ¶56.) As discussed above and for limitation [10.b], the client initiates a lookup of handle-to-pointer mappings (mappings of "identification string[s]," i.e., identifiers, and "location string[s]," i.e., location information, respectively) from a handle server (i.e., location server) storing such mappings, where initiation occurs by sending a request (i.e., location request message) to the handle server to retrieve the location information associated with a digital object. (Section IX.A.1.iii; Ex. 1002, ¶56.)

Third, Kahn discloses "wherein the data is stored at a data repository in the distributed network." As discussed for limitation [10.pre], digital objects (including "data") are stored in a repository ("data repository") in a "distributed network" that includes the repository. (Ex. 1005, Abstract, 6:9-35, 11:16-18, 21:21-24, 73:23-31; Section IX.A.1.i; Ex. 1002, ¶57.)

Fourth, Kahn discloses "a location string associated with the identification string of the data." (Ex. 1002, ¶58.) As discussed above, a handle uniquely identifies a digital object, which includes data ("identification string of the data"), and Kahn further discloses that a pointer ("location string"), which indicates a digital object's location, is associated with a digital object's handle ("identification string of the data"). (Ex. 1005, 28:31-34 ("**A handle for an object may be**

associated with zero or more **object pointers**. **Object pointers contain location information for locating digital objects.**").) Moreover, a handle is used to obtain a list of pointers to the location of a digital object, where each pointer includes the domain name of a repository. (*Id.*, 4:25-29, 6:30-35, 29:8-12.) Additionally, for reasons discussed above, Kahn's pointer discloses the "location string" (or similar terms) as described in the '640 patent. (Ex. 1001, 4:9-11; Ex. 1002, ¶58.)

Fifth, Kahn discloses "wherein a location string...is stored in at least one of a plurality of data location servers." Kahn discloses multiple handle servers and each handle server "holds pointers associated with a subset of all handles." (*Id.*, 37:31-33 ("Handle servers have the following characteristics: a handle server holds pointers associated with a subset of all handles…."); *see also id.*, 7:17-20 ("…multiple handle servers, each handle server having a table which associates handles with pointers."), 21:6-10 ("[M]ultiple servers…provide…handle-to-pointer translation services (called 'handle servers')….").) Because the handle servers store pointers for locating digital objects (including "data"), Kahn's handle servers are the claimed "data location servers." (*Id.*, 7:5-20, 37:31-33; Ex. 1002, ¶59.)

Kahn also discloses the claimed "data location server[]" under the AWS Proceeding's construction of "location server." (Section VIII; Ex. 1001, 4:14-18.) Kahn discloses that each handle server ("location server") is a component attached to a distributed network (i.e., a network-attached component). (Ex. 1005, 7:5-7,

21:21-24, 73:27-31 ("many handle servers [are] distributed on a network").) Each handle server maintains mappings between handles (i.e., identifiers) and pointers (i.e., locations). (Section IX.A.1.i; Ex. 1005, 7:17-20 ("…multiple handle servers, each handle server having a table which associates handles with pointers."), 37:31-33 ("[A] handle server holds pointers associated with a subset of all handles…."), 21:6-10 ("[M]ultiple servers…provide…handle-to-pointer translation services (called 'handle servers')….").) Further, Kahn discloses that the handle server returns identifier/location mappings in response to a request from the client ("first client") for location information associated with a handle (i.e., in response to location request messages from a client). For example, as discussed below for limitation [10.b], pointers (i.e., locations) are returned to the client if the handle server determines it contains the requested location information associated with the handle in the request. (Section IX.A.1.iii.) Given that the returned pointers are associated with the handle in the location information request, a POSITA would have understood that the returned pointers correspond to mappings of the handle to the pointers, consistent with the '640 patent's disclosure. (Ex. 1002, ¶60; Ex. 1001, 2:36-40, 2:60-65, 6:63-67, 16:53-56.) Moreover, Kahn discloses that the full handle record (i.e., including the handle itself and all corresponding data) may be returned in response to a request from the client. (Ex. 1005, 42:1-11 ("[T]he client request may ask for all the data in a handle record…."), 44:16-21.) Thus, a

POSITA would have understood that if pointers are returned to the client from the handle server, the handle itself may also be returned in response to the location request message from the client. (Ex. 1002, ¶60.) Kahn also discloses that identifier/location mappings may be updated/modified in response to a request from the client to add new location information (i.e., another type of location request message from the client). (Ex. 1005, 43:31-44:4, 45:22-46:24; Ex. 1001, 6:46-49.) Thus, Kahn discloses the claimed "data location server[]" under the AWS Proceeding's construction of "location server."

Further, Kahn discloses the claimed "plurality of data location servers" under the AWS Proceeding's construction of "plurality of location servers." (Section VIII.) A POSITA would have understood that the handle servers ("data location server[s]") are arranged in a flat, cluster topology, rather than a hierarchical or tree-like structure where servers can only communicate with nodes immediately above or below it in the tree. (Ex. 1002, ¶61.) Kahn discloses that the hash function is used to distribute handles and associated pointers among the handle servers. (Ex. 1005, 4:19-22, 7:8-9, 37:31-38:3, FIG. 7.) A POSITA would have thus understood that because a hash function is used for distribution across handle servers, Kahn's handle servers are arranged in a non-hierarchical configuration. (Ex. 1002, ¶61.) Indeed, consistent with the above, Patent Owner argued in the AWS Proceeding that the hash function feature requires a non-

hierarchical server structure. (Ex. 1025, 15.) Accordingly, Kahn discloses this limitation.

iii) [10.b]

Kahn discloses this limitation. (Ex. 1002, ¶¶62-63.) **First**, Kahn discloses "the location string associated with the identification string in the data query." As discussed for limitation [10.a], Kahn discloses one or more pointers ("location string") are associated with a handle ("identification string"), where the handle is included in the request ("data query") to retrieve a digital object. (Section IX.A.1.ii; Ex. 1002, ¶62.)

Second, Kahn discloses "transmitting a data location request from the first client to a first data location server to retrieve the location string…, the data location request including the identification string." (Ex. 1002, ¶63.) Kahn discloses that the client ("first client") transmits a handle query in the form of a request pointer information UDP packet ("data location request"), which includes the handle ("identification string"), to the handle server ("first data location server") to retrieve one or more pointers ("location string") for locating the requested digital object. (Ex. 1005, 47:5-11 ("**The [client] system subsequently sends the handle to the handle server as part of a request pointer information UDP packet (step 178).** The handle server then returns its response to the requesting system. If the handle server found the pointers, a list of pointers is

returned (step 180).”); *see also id.*, 7:5-7, 29:5-12, 43:20-23, 44:6-7, 73:32-34; Section IX.A.1.ii.)

iv) [10.c]

Kahn discloses this limitation. (Ex. 1002, ¶¶64-68.) As an initial matter, in the present district court litigation, Patent Owner reads the claimed “redirect message” on a cache miss response due to stale or incorrect location information.[7] (Ex. 1019, ¶¶88-89 (“When the cached location is stale or incorrect, the client detects a miss, which is the redirect message….”).)

First, under Patent Owner’s interpretation, Kahn discloses “if the first data location server does not possess the location string, transmitting a redirect message to the first client.” (Ex. 1002, ¶65.) Kahn discloses transmitting a not-responsible-for-handle message (“redirect message”) to the client (“first client”) in the form of an invalid-handle-server-selected response UDP packet if the handle server, i.e., a “wrong handle server,” does not contain the handle and its associated pointers (“if the first data location server does not possess the location string”). (Ex. 1005,

[7] Petitioner does not acquiesce to Patent Owner’s interpretation of the “redirect message.” Nevertheless, Petitioner explains how Kahn discloses the “redirect message” under Patent Owner’s interpretation. Petitioner also explains how Kahn discloses the “redirect message” if the “redirect message” is interpreted differently.

47:13-15 ("[I]f the handle was sent to the wrong handle server, it returns a **not-responsible-for-handle message** (step 182)."), 73:32-74:4 ("The handle server verifies that the handle falls within the set of handles for [which] it is responsible….**If it is not in this set, …the handle server sends an invalid-handle-server-selected response UDP packet to the requestor**…."), FIG. 8; *see also id.*, 73:1-74:26, FIG. 25.)

Second, under Patent Owner's interpretation, Kahn discloses "the redirect message containing information for use by the first client to calculate a location of a second data location server." (Ex. 1002, ¶66.) If the handle was sent to an incorrect handle server, a not-responsible-for-handle message ("redirect message") is transmitted to the client, informing the client to download an updated table from the handle server directory and recalculate the location of a correct handle server containing the handle ("second data location server"). (Ex. 1005, 47:13-19, 74:5-9.) If a cache miss constitutes a "redirect message," as in Patent Owner's interpretation, Kahn's not-responsible-for-handle message likewise discloses the claimed "redirect message" that contains "information for use by the first client to calculate a location of a second data location server" because it informs the client to download an updated table to determine the location of a correct handle server. (Ex. 1002, ¶66.)

Third, under Patent Owner's interpretation, Kahn discloses "wherein the second data location server contains the location string." (Ex. 1002, ¶67.) Kahn discloses that when attempting to find the correct handle server which contains the pointers ("second data location server contains the location string"), the client reattempts calculating a location of another handle server until it finds the correct handle server.[8] (Ex. 1005, 47:13-18.) A POSITA would have recognized that for any client request for a stored digital object that takes more than one attempt to find a correct handle server, there is some first handle server ("first data location server") that receives a request but does not have the requested pointers and thus sends a redirect message to the client with information the client uses to calculate the location of some second handle server that does have the requested pointers ("second data location server contains the location string"). (Ex. 1002, ¶67.)

Kahn also discloses the claimed "redirect message" even if Patent Owner's interpretation is not applied. (Ex. 1002, ¶68.) As discussed above, the hash

[8] A POSITA would have recognized that the allowed number of attempts would have been set sufficiently large such that if the digital object actually exists in the system, its pointers may be found by repeatedly reattempting the request, i.e., running out of attempts would indicate that the digital object does not exist. (Ex. 1005, 47:18-21; Ex. 1002, ¶67.)

range/handle-server-domain-name table ("redirect message") is downloaded by the client/requesting system ("transmitted…to the first client") in response to client/requesting system receiving a not-responsible-for-handle-message. (Ex. 1005, 47:13-19, 74:5-9.) A POSITA would have understood that when a client downloads the table, the table would be transmitted to the client in the form of a message ("transmitting a redirect message to the first client"). (Ex. 1002, ¶68.) For example, Kahn discloses that the "handle system is based on the UDP protocol" (Ex. 1005, 26:8), and thus a POSITA would have recognized that information transmitted in/by the handle system, e.g., transmitted by the handle server directory of the handle system, as is the case with the transmission of the handle-server-domain-name table, would have been sent as UDP messages. (Ex. 1002, ¶68; *see also* Ex. 1014, 1 ("[User Datagram Protocol] provides a procedure for application programs **to send messages** to other programs with a minimum of protocol mechanism."), Ex. 1016, 4:37-5:12 (incorporating by reference RFC 768 [Ex. 1014]).) Further, as discussed above, this transmission of the Kahn's "redirect message" occurs whenever the not-responsible-for-handle message is sent, i.e., "if the first data location server does not possess the location string." The message, including the handle-server-domain-name table, "contain[s] information for use by the first client to calculate a location of a second data location server." As explained below for limitation [10.d], the transmitted/updated table associates hash

ranges with handle server domain names, and allows the client to determine/calculate the location of the correct handle server ("second data location server") containing a pointer ("location string") for retrieving the digital object associated with the handle. (Section IX.A.1.v; *see also* Section IX.A.1.i.) Thus, Kahn discloses this limitation even if Patent Owner's interpretation is not applied. (Ex. 1002, ¶68.)

v) [10.d]

Kahn discloses this limitation. (Ex. 1002, ¶¶69-70.) For example, Kahn discloses downloading an updated hash range/handle-server-domain-name table and re-applying a hash algorithm/function to a handle to determine/calculate the location of the correct handle server ("calculating the location of a second data location server") at the client ("first client"). As discussed for limitations [10.pre] and [10.c], if a client receives a not-responsible-for-handle message ("redirect message" under Patent Owner's interpretation), the client downloads an updated hash range/handle-server-domain-name table containing hash ranges associated with the locations of handle servers, which contain the handle and its associated pointers. (Sections IX.A.1.i, IX.A.1.iv.) Also as discussed for limitation [10.c], the "redirect message" is also disclosed by Kahn's updated hash range/handle-server-domain-name table being transmitted to the client. (Section IX.A.1.iv.)

The client then repeats the prior steps using the refreshed table. (Section IX.A.1.iv; Ex. 1005, 47:13-19, 74:5-12; Ex. 1002, ¶69.)

This repetition of prior steps includes the client calculating a hash code/value by applying a hash algorithm to the handle at the client, and using the hash code/value to index into the updated table's hash range to determine the domain name/network address of the correct handle server containing the handle and its associated pointer. (Ex. 1005, 38:11-14, 47:1-5, 73:17-20; Section IX.A.3.) Accordingly, Kahn discloses this limitation. (Ex. 1002, ¶70.)

vi) [10.e]

Kahn discloses this limitation. (Ex. 1002, ¶71.) Kahn discloses transmitting, from the client ("first client") to the correct handle server ("second data location server"), the request ("data query") to retrieve a digital object. (Section IX.A.1.ii.) As discussed for limitation [10.d], after receiving a not-responsible-for-handle message ("redirect message" under Patent Owner's interpretation), or alternatively the updated hash range/handle-server-domain-name table itself ("redirect message"), the client determines the location of a correct handle server ("second data location server") using the updated table. (Section IX.A.1.v.) After locating the correct handle server, the request ("data query") to retrieve a digital object is transmitted from the client to the correct handle server. (Ex. 1005, 47:13-18 ("If the handle was sent to the wrong handle server, it returns

a not-responsible-for-handle message (step 182). In this case, **the client system should** download the hash range/handle-server-domain-name table from the handle server directory again and **attempt the mapping again**."); 74:5-26; Section IX.A.1.v.) In particular, the request ("data query") to retrieve the digital object includes sending the handle to the correct handle server ("second data location server") to return an associated pointer to locate and retrieve the digital object. (Ex. 1005, 47:5-11 ("The [client] system subsequently sends the handle to the handle server as part of a request pointer information UDP packet (step 178)."); Ex. 1002, ¶71.)

2. Claim 11

Kahn discloses this limitation. (Ex. 1002, ¶¶72-73.) For example, as discussed for limitations [10.c]-[10.d], Kahn discloses that the result of the client receiving the not-responsible-for-handle-message ("redirect message" under Patent Owner's interpretation) is the subsequent transmission of the updated hash range/handle-server-domain-name table ("data location server table") to the client ("first client"). (Sections IX.A.1.iv-v; Ex. 1005, 47:13-19, 74:5-12.) Thus, transmitting the "redirect message" comprises transmitting the updated table to allow the client to calculate the correct handle server. (Sections IX.A.1.iv-vi; Ex. 1002, ¶72.) As discussed for limitation [10.c], transmitting the updated hash range/handle-server-domain-name table to the client also discloses "transmitting

the redirect message," which comprises "transmitting a data location server table." (Section IX.A.1.iv; Ex. 1002, ¶73.)

3. Claim 12

Kahn discloses this limitation. (Ex. 1002, ¶¶74-75.) **First**, Kahn discloses "wherein calculating the location of the second data location server comprises calculating the location of the second data location server with a function…and based on the data location server table and the identification string." As discussed for limitation [10.d], Kahn discloses that a hash algorithm ("function") is applied by the client ("first client") to the handle ("identification string"), resulting in a hash code/value used to index into the updated hash range/handle-server-domain-name table ("data location server table") to determine/calculate the location of the correct handle server ("calculating the location of the second data location server…and based on the data location server table and the identification string") to obtain one or more pointers associated with the handle. (Section IX.A.1.v; Ex. 1002, ¶74.)

Second, Kahn discloses that the "function [is] commonly known to the first data location server and the first client." (Ex. 1002, ¶75.) As discussed above and for limitation [10.d], the hash algorithm is commonly known to both the client ("first client") and handle servers (including "first data location server"), since both the client and handle servers encounter the hash algorithm during the digital

object retrieval process, e.g., the client generates the hash code for a handle using a hashing algorithm and uses the hash code to index into the hash range/handle-server-domain-name table to determine/calculate the domain name of the handle server. (Section IX.A.1.v; *see also* Sections IX.A.1.i, IX.A.1.iv.) Moreover, the handle servers (including "first data location server") "know[]" about the hash algorithm because each handle server is assigned a range of handles based on hash code/value ranges, which are the result of the application of the hash algorithm to handles. (Section IX.A.1.i; Ex. 1005, 4:19-22 ("distribut[ing] the pointer information with the unique identifier information[] relatively evenly among the servers, using a hashing algorithm"), 29:23-26, 37:31-38:3, 47:1-5, FIG. 7)

4. Claim 13

Kahn discloses this limitation. (Ex. 1002, ¶76.) As discussed for claim 12, the client ("first client") applies a hash algorithm ("hash function") to the handle ("identification string") resulting in a hash code/value used to index into the updated hash range/handle-server-domain-name table ("data location server table") to determine the location of the correct handle server ("obtain the location of the second data location server"). (Section IX.A.3; Ex. 1002, ¶76.) Kahn's teaching of applying a hash function to a handle and the updated table is consistent with the '640 patent's disclosure, where the hash function is applied to the identifier string and the output thereof is used to index into the server table. (Ex. 1001, 14:64-67.)

5. Claim 17

i) [17.pre]

To the extent the preamble is limiting, Kahn discloses the limitations therein. (Ex. 1002, ¶77.) As discussed for limitation [10.pre], Kahn discloses a system for retrieving one or more pointers indicating the location of a digital object stored in a repository of a distributed network ("[a] system for retrieving data location information for data stored in a distributed network"). (Section IX.A.1.i; Ex. 1002, ¶77.)

ii) [17.a]

Kahn discloses this limitation. (Ex. 1002, ¶¶78-79.) **First**, Kahn discloses "a plurality of data repositories configured to store data." For example, Kahn discloses multiple repositories ("a plurality of data repositories") in a network, with each repository storing digital objects (including "data"). (Ex. 1005, 6:22-24, 11:16-18, 12:19-20; Sections IX.A.1.i-ii; Ex. 1002, ¶78.)

Second, Kahn discloses "wherein the data is associated with a respective identifier string in each data repository." (Ex. 1002, ¶79.) Kahn discloses that a digital object (including "data") has an associated transaction record stored in a repository ("data repository"), where the transaction record contains a handle for the digital object ("wherein the data is associated with a respective identifier

string").[9] (Ex. 1005, 14:33-15:7 ("A stored digital object…may have associated with it, in the repository…an associated 'transaction record'…[which] may contain entries such as…the handle….").)

iii) [17.b][10]

Kahn discloses this limitation. (Ex. 1002, ¶¶80-81.) **First**, Kahn discloses "a data location [server] network having a plurality of data location servers, each of the plurality of data location servers containing location strings associated with respective identifier strings." As discussed for limitations [10.pre]-[10.a], Kahn discloses a network including multiple handle servers ("data location [server] network having a plurality of data location servers"). (Ex. 1005, 7:5-20, 21:6-24; Sections IX.A.1.i-ii.) Kahn also discloses each handle server contains one or more pointers ("each of the plurality of data location servers containing location strings") associated with a respective handle ("associated with respective identifier strings"). (Sections IX.A.1.i-iii, IX.A.5.i-ii; Ex. 1002, ¶80.)

[9] Kahn discloses an "identifier string" under the AWS IPR construction of "identifier" for the same reasons as discussed for limitation [10.a]. (Sections VIII, IX.A.1.ii.)

[10] Petitioner assumes that "data location sewer network" is intended to refer to a "data location server network."

Second, Kahn discloses "each of the plurality of data location servers having computer executable code configured to execute the following steps" of limitation [17.c]. For example, Kahn discloses each handle server ("each of the plurality of data location servers") provides handle-to-pointer translation services, i.e., returning a pointer in response to an associated handle, where each handle server can be on one or more computers operating using software. (Ex. 1005, 21:6-15, 26:28-31 ("**software to create and store digital objects and manage rights to the objects**"), 29:5-12, 42:2-11 ("**[E]ach handle server can be implemented as one or more server computers**…."), 44:6-7, 65:14-69:14.) Moreover, Kahn discloses that the handle system interfaces with the handle servers using "routines…written in the C programming language." (*Id.*, 40:31-41:2.) As such, a POSITA would have understood that each handle server ("data location server") provides such services on a computer using "computer executable code" thereof, including the steps recited in limitation [17.c]. (Section IX.A.5.iv; Ex. 1002, ¶81.)

iv) [17.c]

Kahn discloses this limitation. (Ex. 1002, ¶¶82-84.) **First**, Kahn discloses "a data location request from a client to retrieve a location string associated with an identification string provided in the data location request." As discussed for limitation [10.b], a handle query/request pointer information UDP packet ("data location request") is transmitted from a client ("client") to a handle server to

retrieve a pointer ("location string") associated with a handle ("identification string")[11] provided in the handle query/request pointer information UDP packet. (Section IX.A.1.iii ; Ex. 1005, 29:5-12, 47:5-11, 73:32-34; *see also id.*, 7:5-7, 43:20-23, 44:6-7; Ex. 1002, ¶82.)

Second, Kahn discloses "in response to receiving a data location request…transmitting a redirect message to the client if the identification string is not associated with a location string at the data location server." As discussed for limitation [10.c], in response to receiving a handle query/request pointer information UDP packet ("data location request"), a not-responsible-for-handle-message ("redirect message" under Patent Owner's interpretation) is transmitted to the client ("client") if the handle ("identification string") is not associated with a pointer ("location string") at the handle server ("data location server"), i.e., if the request was sent to the wrong handle server. (Section IX.A.1.iv; Ex. 1005, 47:5-15, 73:21-74:4; Ex. 1002, ¶83.)

Third, Kahn discloses "wherein the redirect message contains information for use by the client to calculate a location of a different data location server in the plurality of data location servers, wherein the different data location server

[11] Petitioner assumes that "identification string" is intended to refer to the "identifier string" recited in earlier limitations of claim 17. (Sections IX.A.5.ii-iii.)

contains the location string." (Ex. 1002, ¶84.) As discussed for limitation [10.c], Kahn discloses the not-responsible-for-handle-message ("redirect message" under Patent Owner's interpretation) contains information for the client to use ("information for use by the client") to download an updated table and to determine/calculate the location of a correct handle server containing a pointer locating the digital object ("to calculate a location of a different data location server…wherein the different data location server contains the location string"). (Section IX.A.1.iv.) As discussed for limitation [17.a], each handle server (including "different data location server") is one of multiple handle servers ("a plurality of data location servers"). (Section IX.A.5.ii; *see also* Sections IX.A.1.i-ii.)

6. Claim 18

i) [18.pre]

Kahn discloses this limitation for the same reasons discussed for limitation [17.pre]. (Ex. 1002, ¶85; Section IX.A.5.i.)

ii) [18.a]

Kahn discloses this limitation for reasons discussed for limitation [17.a]. (Ex. 1002, ¶86; Section IX.A.5.ii.)

iii) [18.b]

Kahn discloses this limitation. (Ex. 1002, ¶87.) As discussed for limitations [10.a]-[10.b], Kahn discloses that a client ("client") responds to a request ("responsive to a data query") to retrieve a digital object by sending a handle query/request pointer information UDP packet to a handle server ("to query a data location server") to retrieve one or more pointers ("location information")[12] associated with a handle ("associated with the identifier string").[13] (Sections IX.A.1.ii-iii.)

iv) [18.c]

Kahn discloses this limitation. (Ex. 1002, ¶¶88-90.) **First**, Kahn discloses "a data location server network comprising a plurality of data location servers, at least one of the plurality of data location servers containing location information associated with the identifier string." As discussed for limitation [10.pre], Kahn discloses a network containing multiple handle servers ("a data location server

[12] Kahn discloses "location information" as construed in the AWS Proceeding for the same reasons as discussed for limitation [10.pre]. (Sections VIII, IX.A.1.i.)

[13] Kahn discloses an "identifier string" under the AWS IPR construction of "identifier" for the same reasons as discussed for limitation [10.a]. (Sections VIII, IX.A.1.ii.)

network comprising a plurality of data location servers"), where each handle server ("at least one of the plurality of data location servers") contains one or more pointers ("location information") associated with a handle ("identifier string"). (Section IX.A.1.i; Ex. 1005, 7:5-20, 21:6-24; Ex. 1002, ¶88.)

Second, Kahn discloses "wherein each of the plurality of data location servers comprises computer executable code configured to execute the following steps" of limitations [18.d] and [18.e]. (Sections IX.A.6.v-vi.) As discussed for limitations [17.b]-[17.c], each handle server ("each of the plurality of data location servers") provides handle-to-pointer translation services on a computer using "computer executable code" thereof, including the steps recited in limitation [18.d] and [18.e]. (Sections IX.A.5.iii-iv, IX.A.6.v-vi; Ex. 1002, ¶89.)

As discussed for limitation [10.b] and further below, Kahn discloses that the steps of limitations [18.d]-[18.e] are performed "in response to receiving a data location request from the client." (Sections IX.A.1.iii, IX.A.6.v-vi.) Kahn discloses in response to receiving a handle query/request pointer information UDP packet ("data location request") the client transmitting a handle to a handle server to retrieve a pointer to the location of a digital object. (Ex. 1005, 29:5-12, 47:5-8; *see also id.*, 7:5-7, 43:20-23, 44:6-7; Section IX.A.6.iii (discussion of handle servers); Ex. 1002, ¶90.) Moreover, consistent with as discussed for limitations [10.c]-[10.d], each handle server ("data location server") performs processes to

determine if it contains a particular handle and its associated pointer ("location string"), as further explained below for limitations [18.d] (discussing actions performed if the claimed "data location server" does contain the location string) and [18.e] (discussing actions performed if the claimed "data location server" does not contain the location string). (Sections IX.A.1.iv-v; *infra* Sections IX.A.6.v-vi; Ex. 1002, ¶90.)

v) [18.d]

Kahn discloses this limitation. (Ex. 1002, ¶¶91-92.) **First**, Kahn discloses "the data location server contains the location string associated with the identification string provided in the data location request." For example, Kahn discloses a handle server ("data location server") containing one or more pointers ("location string") associated with a handle ("identification string")[14] provided in the handle query/request pointer information UDP packet ("data location request"). As discussed for limitations [10.pre] and [10.b], Kahn discloses the client transmitting a handle in a handle query or request packet to the handle server to determine if the handle server has a pointer associated with the digital object's

[14] Petitioner assumes that "identification string" is intended to refer to the "identifier string" recited in earlier limitations of claim 18. (Sections IX.A.6.ii-iv.)

handle. (Ex. 1005, 29:5-12, 47:5-8; *see also id.*, 7:5-7, 43:20-23, 44:6-7; Sections IX.A.1.i, IX.A.1.iii; Ex. 1002, ¶91.)

Second, Kahn discloses "if the data location server contains the location string…the data location server transmits location information for use by the client to calculate a location of the data associated with the identification string." Kahn discloses that if a handle server contains the queried handle and its associated one or more pointers ("if the data location server contains the location string"), the handle server returns the pointers associated with the handle ("the data location server transmits location information") to the client for the client to determine ("for use by the client to calculate") the location of the digital object ("a location of the data") associated with the handle ("identification string"). (Sections IX.A.1.i-iii.) Kahn discloses "[i]f the handle server is responsible for the handle [and its associated pointer], it verifies that the handle is present in its database…. For each pointer returned by the handle server, the requesting [client] system uses the pointer to attempt to obtain a copy of the digital object." (Ex. 1005, 74:13-26; *see also id.*, 47:9-11, 73:21-27, FIG. 8 (step 180); Sections IX.A.1.i-iii.) Kahn discloses that the returned pointers include the "domain name of [the] storage system (repository)" of the requested digital object. (Ex. 1005, 29:8-12.) Thus, a POSITA would have understood that to retrieve the digital object, the client uses the information included in the pointers ("location information for use by the

client"), e.g., the repository domain name, to determine/calculate the location of the digital object (including "data") associated with the requested handle ("to calculate a location of the data associated with the identification string"). (Ex. 1002, ¶92; *see also* Ex. 1005, 44:30-32 ("Repository names are not actual network addresses; they must first be mapped to network locations.").)

vi) [18.e]

Kahn discloses this limitation. (Ex. 1002, ¶¶93-94.) **First**, Kahn discloses "wherein the redirect message contains redirect information for use by the client to calculate a location of a different data location server in the plurality of data location servers, wherein the different data location server contains the location string." As discussed for limitation [10.c], Kahn discloses that a handle server transmits a not-responsible-for-handle message ("redirect message" under Patent Owner's interpretation) to the client ("client"), where the message informs the client to download an updated hash range/handle-server-domain-name table ("wherein the redirect message contains redirect information for use by the client"), for use by the client in determining/calculating a correct handle server containing a pointer to the location of the digital object ("to calculate a location of a different data location server in the plurality of data location servers, wherein the different data location server contains the location string"). (*Supra* Section IX.A.1.iv; Ex. 1002, ¶93.)

Second, Kahn discloses "if the data location server does not contain the location string associated with the identification string, the location server transmits a redirect message to the client." (Ex. 1002, ¶94.) As discussed for limitation [10.c], Kahn discloses that if a handle server ("data location server") does not contain the handle ("identification string") associated with one or more pointers ("location string"), the handle server transmits a not-responsible-for-handle message ("redirect message" under Patent Owner's interpretation) to the client ("client"). (Section IX.A.1.iv; Ex. 1005, FIG. 8 (step 182), 47:13-15, 73:32-74:4.)

7. Claim 19

Kahn discloses this limitation. (Ex. 1002, ¶95.) As discussed for limitation [10.d], claim 12, and claim 13, Kahn discloses a hash algorithm ("function") commonly known to the client ("client") and multiple handle servers ("plurality of data location servers"), as the client and handle servers either apply or are associated with the hash algorithm while performing the digital object retrieval process ("wherein the client and the plurality of data location servers each comprise a function commonly known to the client and the plurality of data location servers"). (Sections IX.A.1.v, IX.A.3, IX.A.4.) As further discussed therein, the client ("client") applies the hash algorithm to a handle, resulting in a hash code/value used to index into the hash range/handle-server-domain-name

table ("redirect information") to determine the domain name of the correct handle server containing a handle and its associated pointer to the location a digital object ("wherein the client is configured to apply the commonly known function to the location information or redirect information"). (Sections IX.A.1.v, IX.A.3, IX.A.4.)

8. Claim 24

Kahn discloses this limitation. (Ex. 1002, ¶96.) Kahn discloses that all handles are partitioned amongst multiple handle servers, where the handles are partitioned/distributed based on the hash algorithm and hash range/handle-server-domain-name table, such that each handle server is assigned a portion of the handle space. (Ex. 1005, 7:5-9 ("Multiple [handle] servers…each serv[e] a portion of the handle space."), 37:31-38:3 ("[T]he set of all hash values is partitioned among the set of all handle servers."), 46:22-24 ("[T]he handle server directory…holds a table…which associates hash ranges…with domain names of handle servers…."), 47:1-5.) Thus, handles and their associated pointers ("location information"), which are stored in handle/pointer mappings on the handle servers, are likewise partitioned/distributed over multiple handle servers according to the hash range/handle-server-domain-name table, such that the pointers on a given handle server represent only a portion of the full set of pointers. (Ex. 1002, ¶96.) A POSITA would have therefore understood that the pointers ("location

information") on a given handle server comprise a portion of the full handle-server-domain-name table ("hash table") distributed over the plurality of handle servers ("comprises a portion of a hash table distributed over the plurality of data location servers") because the pointers are associated with a portion of the handles in the table. (Ex. 1002, ¶96.)

B. Ground 2: Kahn in View of Vingralek Renders Claims 10-24 Obvious

1. Claim 10

Kahn in view of Vingralek discloses and/or suggests claim 10. (Ex. 1002, ¶¶97-106.) As discussed for Ground 1, Kahn discloses claim 10. (Section IX.A.1.) To the extent that limitation [10.c] is interpreted to require that the "redirect message" is sent from a "data location server" and/or to the extent Kahn's "redirect message" does not disclose the redirect message "containing information for use by the first client to calculate a location of a second data location server," a POSITA would have found it obvious to implement these features in view of Vingralek. (Ex. 1002, ¶97.)

Like Kahn, Vingralek relates to storing data in a distributed network. For example, Vingralek discloses a "distributed file organization system" where a file is "organized into buckets…spread across multiple servers" (Ex. 1007, 4) and clients can modify the file's distribution by issuing requests to insert/delete/retrieve records for a given key *K* (akin to an identifier of a record) in the file, such that the

file grows or shrinks dynamically (*id.*, 9). Vingralek offers "a new approach for hash-based distributed files that strives for…'cost-conscious' scalability…to minimize the number of servers…used for providing the necessary performance capacity." (*Id.*, 6.) Clients locate servers for insertion/update/deletion, and perform "key-based retrieval through a hash function whose target domain is dynamically expanded as buckets are split and servers are added, based on the principle of linear hashing." (*Id.*, 5, 8.) Vingralek's hash-based distributed file system is implemented using an address table, which a POSITA would have understood is akin to a hash table, to look up the server that corresponds to a hashed key. (*Id.*, 4, 6.) Thus, Vingralek is similar to Kahn, which discloses a hash-based object distribution system and using a handle to locate/retrieve data. (Section IX.A.1.i.). A POSITA would have therefore been motivated to consider the teachings of Vingralek when implementing Kahn's system/method. (Ex. 1002, ¶98.)

Vingralek additionally discloses propagating updates to clients' and other servers' address tables and hash functions in a deferred manner. (Ex. 1007, 6.) To limit communication overhead, clients learn about changes to the hash function and associated address table only when a client request reaches a wrong server, in which case the server forwards the client's request to the correct server and provides updates to the client's hash function and address table. (*Id.*, 5-6; Ex.

1002, ¶99.) Thus, Vingralek discloses updates to a client's stale hash function and address table are deferred until a wrong first server is reached and the location of a second server needs to be calculated.

Vingralek's method of deferred hash function and address table updates operates as follows. A hash function is applied to a key, which is used as an index to an address table listing available servers. (Ex. 1007, 9.) Each client and server has its own copy of the address table, but each table may have stale information if changes have been made to the servers since the table was last updated. (*Id.*, 10 ("The client[] and…server[]…may have stale information on…the address table….").) Similarly, each client and server may have stale information on the hash function. (*Id.*, 5 ("client's stale information on the hash function"), 9 ("file level" is a parameter derived from the index to the hash function), 10 ("The client[] and…server[]…may have stale information on the file level….").) When a client seeks to retrieve records associated with a given key, it applies its current hash function to the key, identifies the server where that key is located by using the hashed key as an index into the current copy of the address table, and then sends the key to that first server. (*Id.*, 9-10.) The first server that receives the key also applies its hash function to the key and identifies in its copy of the address table the server where the key resides. (*Id.*, 10-11.) If the key was indeed sent to the correct server, then that server performs the requested operation. (*Id.*, 11.) But if

the key was sent to the wrong server, the first server forwards the request to another server it has identified through applying the hash function, and sends a message to the client to update the client's hash function and address table. (*Id.*; *see also id.*, 25 ("client requests and server responses" are "messages"); Ex. 1002, ¶100.)

The key is eventually received by the correct server and the request from the client is executed. (Ex. 1007, 11.) Via the address table update, "clients gradually bring their perception of the file up-to-date, so that the probability of wrongly addressed messages is significantly reduced." (*Id.*) Vingralek explains that if a server undergoes changes (e.g., if information is moved to another server), the server receives a new address table and the parameters for the hash function are recalculated. (*Id.*, 12-13.) Consistent with the above, a POSITA would have understood that a server would have up-to-date information with respect to its own keys/information, such that the process of updating the client's address table and hash function each time a request is forwarded to the next server (until the request reaches a correct server containing the requested information, i.e., the "second data location server") would have gradually brought the client's perception of the address table and hash function up-to-date. (Ex. 1002, ¶101.) Thus, Vingralek discloses that if a first server does not have the requested information, a message ("redirect message") is sent to the client with information to update the client's

hash function and address/hash table, which the client uses when calculating the location of a server. (Ex. 1002, ¶101.)

Additionally, Vingralek discloses that the "most salient feature" of its approach, where a client's hash function and table are updated in a deferred manner, "is that its communication overhead is largely independent of the number of servers and clients in the system; thus, it is considered a scalable approach." (Ex. 1007, 5.) Thus, Vingralek discloses implementing a redirection mechanism and additionally includes improved features to minimize the number of servers and reduce the cost of scaling up a distributed system. (*Id.*, 6, 8; Ex. 1002, ¶102.)

A POSITA would have been motivated to include features similar to those disclosed by Vingralek when implementing Kahn by providing a deferred hash function and table update to the client (for the client to calculate a location of a correct handle server). (Ex. 1002, ¶103.) Like Vingralek, Kahn discloses that both client and handle server store copies of the hash table. (Ex. 1005, 38:9-17 ("**[E]ach handle server will contain such a directory [server table]). This table will generally be downloaded by each participating site** frequently enough to be 'acceptably' up-to-date at all times."), 46:29-32 ("**[A] client system downloads the table that associates hash ranges with handle server domain names from the handle server directory** for future use.").) Further, Kahn discloses its data/file distribution is subject to change, e.g., when the data and/or the data

repository have changed. (*Id*., 20:29-33, 33:3-14, 45:10-12.) Accordingly, Kahn's hash table and hash function are also subject to change. (Ex. 1002, ¶103; Ex. 1005, 46:22-34 ("**frequent changes in the table** may necessitate [the client downloading the table] every time."), 74:5-9 ("If the requesting system receives an invalid-handle-server-selected response UDP packet, it **refreshes its hash code/handle server table** from the handle server directory.").) Thus, consistent with Vingralek, Kahn's system would have benefited from implementing such a hash table and hash function update, as it would have allowed the client to use a correct/updated hash table and hash function to locate a handle server for data retrieval. (Ex. 1002, ¶103.) Further, Vingralek explains that this approach would have decoupled the communication overhead from the number of servers/clients in the system and improved scalability, consistent with Kahn's goal and expectation of scaling. (Ex. 1005, 9:23-27 ("**The handles…are designed to be extensible and expandable to accommodate virtually any number of objects over many years.**"), 33:3-14 ("[T]he identifier's namespace must be variable and unrestricted, **since the number of digital objects created may be expected to increase**…."); Ex. 1007, 5.) Thus, a POSITA would have been motivated to include deferred hash table and hash function updates from the handle server, similar to that of Vingralek, when implementing Kahn's system/process, so as to reduce the communication overhead associated with Kahn's existing method of updating the

client's hash table and function (i.e., frequent downloads from the handle directory server). (Ex. 1002, ¶103; Ex. 1005, 46:22-34.)

A POSITA would have understood that such a modification would have further reduced communication overhead because as discussed for limitation [10.c], Kahn, similar to Vingralek, also uses a deferred update process via message sent from handle server to client, where the transmission of a not-responsible-for-handle message from a handle server results in the client's download of an updated hash range/handle-server-domain-name table for determining/calculating the location of a correct handle server after reaching a wrong handle server. (Section IX.A.1.iv; Ex. 1002, ¶104.) Combining the transmission of the message from the handle server with the update to the client's hash function/table would have reduced the number of required interactions between system components, thereby reducing communication overhead. (Ex. 1002, ¶104.)

Additionally, a POSITA would have been motivated to implement the deferred hash function/table updates in a system similar to Kahn, e.g., by modifying Kahn's mechanism of notifying the client. As disclosed in Kahn, the client ("first client") would apply its current hash function to its downloaded copy of the hash table, and send a request to the first handle server ("first data location server"). (Ex. 1002, ¶105.) While Kahn discloses that "[t]he handle server verifies that the handle falls within the set of handles for whose hash values it is

responsible," it does not explicitly disclose the specifics of its verification method. (Ex. 1005, 73:32-34.) Thus, a POSITA would have been motivated to use a verification method similar to that of Vingralek, i.e., applying the hash function to the server's own hash table, as discussed above, especially because Kahn already discloses that the handle server stores a copy of the hash table. (Ex. 1002, ¶105; Ex. 1005, 38:9-17, 46:29-31.) Thus, in the Kahn-Vingralek combination, for example, the first handle server would apply the hash function to the handle and use the result as an index into the hash table as a means of determining whether the handle server has the requested location information. (Ex. 1002, ¶105.) If the first handle server does not have the requested location information, i.e., the pointers associated with the handle ("if the first data location server does not possess the location string"), the handle server would transmit a message ("redirect message") to the client, including the handle server's hash table and hash function for use by the client to calculate a location of a second handle server that contains the pointers associated with the handle ("transmitting a redirect message to the first client, the redirect message containing information for use by the first client to calculate a location of a second data location server, wherein the second data location server contains the location string"). (*Id.*) The client would then calculate the location of the correct handle server ("calculating the location of the second data location server at the first client") and send the request to that correct handle server

("transmitting the data query from the first client to the second data location server"), as discussed for Ground 1, limitations [10.d]-[10.e]. (Sections IX.A.1.v-vi.)

A POSITA would have understood that the above-described implementation, where the client calculates a correct handle server and forwards the request to that server (e.g., as in Kahn), would have been one of only several options, i.e., calculating and forwarding at either the client or the handle server that processes the information request. Thus, transmitting a redirect message with information *for the client* ("first client") to calculate the location of a second handle server would have been obvious because it would have been one of a "finite number of identified, predictable solutions." *Perfect Web Techs., Inc. v. InfoUSA, Inc*., 587 F.3d 1324, 1331 (Fed. Cir. 2009). A POSITA would have further understood that implementing such a redirect message would allow hash table/function updates to propagate from handle server to client, while reducing the number of updates, e.g., downloads performed from the handle directory servers (i.e., by reducing the number of downloads by the client). (Ex. 1002, ¶105.)

A POSITA would have also had a reasonable expectation of success combining the teachings of Kahn and Vingralek. (Ex. 1002, ¶106.) Both references are in the same field of art and both relate to hash-based distributed computing/file systems. (*See supra*.) Moreover, as discussed above, Kahn already

utilizes a message to provide updated information to the client when a wrong server is reached, but would benefit from reducing communication overhead by implementing Vingralek's teaching. A POSITA would have also had the skill to implement and reasonable expectation of success in achieving such a modification, because it would have involved a combination of known technologies (e.g., known distributed data storage systems (Kahn)) according to known methods (e.g., known methods of deferred updates for hash functions and tables using a redirect message (Vingralek)) to yield the predictable result of a distributed data storage system that limits communication overhead and improves scalability, as discussed above. *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 416 (2007). Accordingly, Kahn in view of Vingralek discloses and/or suggests this limitation. (Ex. 1002, ¶106.)

2. Claim 11

The Kahn-Vingralek combination discloses and/or suggests this limitation. (Ex. 1002, ¶107.) As discussed for Ground 2, claim 10, a first handle server that does not have the requested location information would send a message ("transmitting the redirect message") to the client ("first client") to update the client's hash function and hash table. (Section IX.B.1.) A POSITA would have understood that, to update the hash table, the redirect message could include either the specific information to be updated or the entire table for replacing the table, depending on the extent of the update required. (Ex. 1002, ¶107.) Thus,

transmitting an entire hash table ("data location server table") to the client ("first client") would have been obvious because it would have been one of a "finite number of identified, predictable solutions." *Perfect Web Techs.*, 587 F.3d at 1331. Indeed, Vingralek suggests that the entire hash table may be sent to the client to update a stale table. (Ex. 1007, 11 ("[I]f [key] *K* has been sent to the wrong server, the server…updates the client's view with…its address table.").)

3. Claim 12

The Kahn-Vingralek combination discloses and/or suggests this limitation. (Ex. 1002, ¶108.) As discussed for Ground 2, claim 11, the redirect message contains a hash table ("data location server table") that the client ("first client") uses to calculate the location of a correct handle server ("calculating the location of the second data location server") using a hash function. (Section IX.B.2.) As discussed for Ground 2, claim 10, both client and handle servers (including "first data location server") apply the function in the retrieval process (the client to determine where to send a request and the handle server to determine if it has pointers associated with a handle), and updates to the function are shared among the client and handle servers so each component has an up-to-date perception of the function ("a function commonly known to the first data location server and the first client"). (Section IX.B.1.) As further discussed therein, the updated hash function is applied by the client to the newly updated hash table and the handle

("based on the data location server table and the identification string") to determine the location of the second handle server ("calculating the location of the second data location server"). (*Id*.)

4. Claim 13

The Kahn-Vingralek combination discloses and/or suggests this limitation. (Ex. 1002, ¶109.) As discussed for Ground 2, claim 12, the disclosed "function" comprises a hash function. (Section IX.B.3.) Moreover, as discussed therein, the client ("first client") applies the updated hash function to the handle ("identification string") and updated hash table ("data location server table") to determine the location of the second handle server ("obtain the location of the second data location server"). (*Id.*; Section IX.A.4; Ex. 1002, ¶109.)

5. Claim 14

The Kahn-Vingralek combination discloses and/or suggests this limitation. (Ex. 1002, ¶110.) As discussed for Ground 2, claim 10, sending a message ("transmitting the redirect message") includes an updated hash function ("transmitting a function") and hash table sent to the client ("first client") to update the client's hash function and hash table. (Section IX.B.1.)

6. Claim 15

The Kahn-Vingralek combination discloses and/or suggests this limitation. (Ex. 1002, ¶111.) As discussed for Ground 2, claim 10, the client calculates a

location of a correct handle server ("calculating at the first client the location of the second data location server") by applying the transmitted/updated hash function ("transmitted function") to the handle ("identifier string").[15] (Section IX.B.1.)

7. Claim 16

The Kahn-Vingralek combination discloses and/or suggests this limitation. (Ex. 1002, ¶112.) As discussed for Ground 1, limitation [10.d], Kahn discloses applying the hash algorithm to the handle for indexing into the hash range/handle-server-domain-name table to generate the domain name/network address of the correct handle server ("URL of the second data location server"). (Section IX.A.1.v; Ex. 1005, 38:11-14, 46:22-47:5.) A POSITA would have understood that applying the transmitted/updated hash function ("transmitted function") to the handle, as discussed for claim 15, would have likewise generated the domain name/network address of the correct handle server ("URL of the second data location server"). (Section IX.B.6; Ex. 1002, ¶112.) Moreover, Kahn discloses that its system, which includes a handle server ("data location server"), may be used on a network such as the Internet. (Ex. 1005, 2:25-27 ("digital objects available on **networks (e.g., Internet)**"), 19:31-20:2 ("**broader network**

[15] Petitioner assumes the "identifier string" is intended to refer to the "identification string" recited claim 10. (Sections IX.A.1.ii-iii.)

structure (e.g., the Internet)"); *see also id.*, 7:5-7 ("**handle server, accessible on the network**").) A POSITA would have understood that a network address on the Internet is a URL, and thus the location of any server connected by the Internet (e.g., the handle servers) is defined by such a URL. (Ex. 1002, ¶112.)

8. Claim 17

The Kahn-Vingralek combination discloses and/or suggests this limitation. (Ex. 1002, ¶¶113-114.) As discussed for Ground 1, Kahn discloses claim 17. (Section IX.A.5.) To the extent that Kahn's redirect message described for limitation [17.c] does not disclose that the redirect message contains information for the client for use to calculate the location of a different location server, a POSITA would have been motivated and found it obvious to implement this feature in view of Vingralek for reasons similar to those as discussed for claim 10 in Ground 2. (Section IX.B.1; Ex. 1002, ¶113.)

For example, as discussed for claim 10 in Ground 2, the Kahn-Vingralek combination discloses and/or suggests that, after the first handle server receives the request from the client ("in response to receiving a data location request from a client") to retrieve pointers associated with a handle provided in the request ("to retrieve a location string associated with an identification string provided in the data location request"), the handle server applies the hash function to the handle and to its own copy of the hash table to determine whether it has the requested

pointers ("location string") associated with the handle ("identification string"). (Section IX.B.1.) As further discussed, if the first handle server does not have the requested pointers associated with the handle ("if the identification string is not associated with a location string at the data location server"), the handle server would send a message to the client ("transmitting a redirect message") to update the client's hash function and hash table, thus allowing the client to use the updated hash function to calculate the correct handle server for sending the request ("wherein the redirect message contains information for use by the client to calculate a location of a different data location server in the plurality of data location servers, wherein the different data location server contains the location string"). (*Id*.; Ex. 1002, ¶114.)

9. Claim 18

The Kahn-Vingralek combination discloses and/or suggests this limitation. (Ex. 1002, ¶¶115-116.) As discussed for Ground 1, Kahn discloses claim 18. (Section IX.A.6.) To the extent that Kahn's redirect message described for limitation [18.e] does not disclose that "the redirect message contains redirect information for use by the client to calculate a location of a different data location server," a POSITA would have been motivated and found it obvious to implement this feature in view of Vingralek. (Section IX.B.1; Ex. 1002, ¶115.)

As discussed for Ground 2, claim 10, in the Kahn-Vingralek combination, after the first handle server receives the request, it applies the hash function to the handle and its copy of the hash table to determine whether it has the requested pointers ("location string"). (Section IX.B.1.) If the server does not have the requested pointers associated with the handle ("if the data location server does not contain the location string associated with the identification string"), it would send a message to the client("the location server transmits a redirect message to the client") containing an updated hash function and hash table ("redirect information") for updating the client's hash function and hash table. (*Id*.) This step would allow the client to use the updated hash function and hash table to calculate the location of the correct handle server in the system of multiple handle servers ("to calculate a location of a different data location server in the plurality of data location servers") to send the request to, where the "correct" handle server contains the requested pointers ("wherein the different data location server contains the location string"). (Section IX.B.1; Ex. 1002, ¶116.)

10. Claim 19

The Kahn-Vingralek combination discloses and/or suggests this limitation. (Ex. 1002, ¶117.) As discussed for Ground 2, claim 12, a hash function is known to both the client and the handle servers, since both apply the function in the retrieval process and updates to the function are shared among client and handle

servers ("wherein the client and the plurality of data location servers each comprise a function commonly known to the client and the plurality of data location servers"). (Section IX.B.3.) As discussed for Ground 2, claim 11, the redirect message contains a hash table and hash function ("redirect information") that the client ("client") uses to determine a correct handle server by applying the hash function to the handle and hash table ("wherein the client is configured to apply the commonly known function to the…redirect information"). (Section IX.B.2.)

11. Claim 20

The Kahn-Vingralek combination discloses and/or suggests this limitation for the reasons given for Ground 2, claim 11. (Ex. 1002, ¶118; Section IX.B.2.)

12. Claim 21

The Kahn-Vingralek combination discloses and/or suggests this limitation. (Ex. 1002, ¶119.) As discussed for Ground 2, claim 19, the hash function known to the client and handle servers ("commonly known function comprises a hash function") is applied by the client ("client") to the handle ("identification string") and hash table ("data location server table") to determine the location of a second handle server ("obtain the location of the different data location server"). (Section IX.B.10.)

13. Claim 22

The Kahn-Vingralek combination discloses and/or suggests this limitation. (Ex. 1002, ¶120.) As discussed for claims 14 and 15, the handle server transmits a message ("redirect message") that includes an updated hash function ("transmitted function") for use by the client ("client") in calculating the location of the correct handle server. (Section IX.B.5, IX.B.6.)

14. Claim 23

The Kahn-Vingralek combination discloses and/or suggests this limitation. (Ex. 1002, ¶121.) As discussed for claim 15, the client calculates the location of another handle server ("wherein the client is configured to calculate the location of the different data location server") by applying the hash function transmitted in the redirect message to the handle ("by applying the transmitted function to the identifier string"). (Section IX.B.6.)

15. Claim 24

Kahn discloses this limitation for the reasons given for Ground 1, claim 24. (Ex. 1002, ¶122; Section IX.A.8.)

X. DISCRETIONARY DENIAL IS NOT APPROPRIATE

A. The Board Should Not Deny Institution Under § 325

Discretionary denial under § 325(d) is not appropriate here because this petition does not present the same or substantially the same prior art or arguments previously presented to the Office during prosecution. (Ex. 1004; Ex. 1001,

Cover.) *Advanced Bionics, LLC v. Med-El Elektromedizinische Geräte GMBH,* IPR2019-01469, Paper 6 at 8 (Feb. 13, 2020) (precedential). Moreover, there are material differences between the art/arguments here and those during prosecution (*see, e.g.*, Ex. 1004, 442-449, 492-498, 142, 150, 184, 190), including features believed to be missing (*id.*, 458-461, 521-526). Discretionary denial is also not warranted in view of the completed *ex parte* reexamination proceedings, since this petition does not present the same or substantially the same prior art or arguments presented in those reexams, and there are material differences between the art/arguments here with those presented therein (*see*, *e.g.*, Ex. 1026, 21-24; Ex. 1028, 11), including features believed to be missing (Ex. 1027, 21 ("Jaeger…does not appear to teach a redirect message that is sent to a client"), 31 ("None of the [references] are seen as teaching a redirect message"), 36 ("Combining the [references] still fails to provide a teaching of redirect message"); Ex. 1030, 6 ("The cited prior art fail to teach individually or in combination: '…transmitting a redirect message to the client…'")). And the Office erred in a material manner by allowing the claims despite such features being known. *Advanced Bionics*, Paper 6 at 10-11; *Becton, Dickinson*, Paper 8.

Discretionary denial is also not warranted in view of the pending *ex parte* reexamination initiated by a different party, AWS. (Ex. 1032; Section II.) First,

Petitioner had no involvement with the reexamination, and Petitioner developed the present petition independently of AWS.

Second, this petition and the reexamination are based on materially different grounds and positions. None of the positions in the reexamination relies on Kahn, which is asserted in each of these grounds. (Ex. 1032, 6-7, 16-20; Sections IV, IX.) Thus, there is no substantial overlap between the grounds here and those remaining in the pending EPR.

Third, the pending claims of the reexamination and the challenged claims in this IPR are not coextensive. (Sections II, IV; Ex. 1032, 20.)

Fourth, denying institution in view of a third-party reexamination would be unfairly prejudicial. The reexamination does not challenge all of the claims that are challenged in this proceeding. Moreover, it is Patent Owner that decided to accuse separate companies of infringement, each with separate interests in challenging the patent. And, given the statutory deadline to file an IPR, Google could not have waited until the conclusion of the reexamination. *See Mueller Sys., LLC v. Rein Tech, Inc.*, IPR2020-00100, Paper 9 at 12-14 (P.T.A.B. May 12, 2020) ("Petitioner would be barred from challenging claims…following the Reexamination Proceeding.").

Fifth, the reexamination remains at an early stage, which counsels against discretionary denial. *See Mueller* at 13. Accordingly, discretionary denial is not

warranted in view of the EPR. *See Pfizer Inc. v. Hoffman-LA Roche Inc.*, IPR2018-01219, Paper 12 at 5-10 (P.T.A.B. Dec. 18, 2018) (instituting IPR where "the grounds differ" from those in the pending EPR).

B. Discretionary Denial Under *General Plastic* Is Not Appropriate

Petitioner has never before filed a Patent Office challenge to the '640 patent. Patent Owner may argue that the '640 patent is already at issue in proceedings initiated by different parties. (Section II.) The facts here, however, do not support a discretionary denial based on those proceedings.

The Board has set forth seven factors in determining whether to exercise discretion to deny a petition under 35 U.S.C. § 314(a) due to an earlier proceeding involving the same patent. *Gen. Plastic Indus. Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper 19 at 15-19 (Sept. 6, 2017) (precedential) ("*General Plastic*"). The *General Plastic* factors confirm that this Petition should be considered on the merits.

With respect to the first factor, Petitioner is not a party, real party-in-interest, or privy to the other Patent Office proceedings and this is Petitioner's first challenge to the '640 patent, which "weighs especially heavily against a discretionary denial." *See Unified Patents, Inc. v. Certified Measurement, LLC*, IPR2018-00548, Paper 7 at 7-8 (PTAB Sept. 5, 2018). Moreover, unlike *Valve*, AWS and Petitioner were sued at different times—four and a half years apart—and

the AWS-challenged claims (Section II) and the challenged claims in this IPR are not coextensive. *Valve Corp. v. Elec. Scripting Prods., Inc.*, IPR2019-00062, Paper 11 at 10 (PTAB April 2, 2019) (precedential).

The second to fifth factors "bear little relevance" here because Petitioner has never before challenged the '640 patent. *Unified* at 7-8 ("Once resolution of factor 1 indicates that Petitioner had not previously filed a petition against the same patent, factors 2-5 bear little relevance….") (internal citation omitted). Petitioner has no relationship to the earlier challenges to the '640 patent. And there is no indication that AWS was aware of the Kahn-based prior art combinations at issue here, which disclose and/or suggest many of the challenged claims. In fact, the district court litigation against Petitioner was filed in June 2023, several months after AWS filed its currently-pending reexamination request in February 2023. (Ex. 1031, 234.)

As to the sixth factor, while the Board has finite resources, instituting this petition would be no more a burden on these finite resources than instituting any other petition. And for the seventh factor, there is no readily identifiable roadblock for the Board to issue a final determination within the statutory one-year limit.

C. The Board Should Not Deny Institution Under *Fintiv*

The merits of Petitioner's arguments are compelling and the evidence in support is substantial. That "alone demonstrates that the PTAB should not

discretionarily deny institution under *Fintiv*." (Ex. 1036, 4-5.) In any event, the six *Fintiv* factors do not justify denying institution.

The **first factor** (stay) weighs strongly against denial. Although Google has not yet moved to stay in district court pending IPR, the Northern District of Illinois has stayed a number of cases pending IPR proceedings since 2020. (Ex. 1033, 1, 2, 5, 6, 10.)

The **second factor** (trial proximity) weighs strongly against denial, since "the median time-to-trial is…after the projected…final written decision." (Ex. 1036, 9.) The median time-to-trial in the Northern District of Illinois is close to 57 months. (Ex. 1035, 47.) The district court case was filed on June 30, 2023, so trial would occur in April 2028. On the other hand, with this petition filed in June 2024, a final written decision may be expected around December 2025. Moreover, the Board has relied on various justifications, such as diligence. *Verizon Business Network Servs., Inc. v. Huawei Techs. Co*., IPR2020-01141, Paper 12 (Jan. 14, 2021). This petition was diligently filed shortly after Patent Owner served its infringement contentions identifying asserted claims on April 25, 2024. (Ex. 1034, 1.)

The **third factor** (investment in parallel proceedings) also weighs strongly against denial. The district court case is in the early stages. (*Id*., 1.) The date of claim construction hearing is still to be determined but is likely to be no earlier

than April/May 2025. (*Id.*, 2.) Moreover, the district court's discovery process is currently stayed pending resolution of Google's Motion to Dismiss until July 15, 2024. (Ex. 1052, 1; Ex. 1055, 1.)

The **fourth factor** (overlap) also weighs against denial, since Petitioner challenges more claims than Patent Owner may ultimately assert in its final contentions in the district court litigation, i.e., 10 claims per patent and 20 claims overall from the '640, '170, and '978 patents. (Ex. 1053, 10.). Thus, there is no complete overlap of claims/issues between the present proceeding and the district court litigation. The Board has found that reduced issues in district court litigation, including the reduced number of asserted claims, "tilt[] in favor of institution." *Apple, Inc. v. SEVEN Networks, LLC*, IPR2020-00156, Paper 10 at 17, 21-22 (P.T.A.B. June 15, 2020). Moreover, institution here will serve the "interest of overall system efficiency and integrity" because it would allow this proceeding to "continue in the event" the district court litigation "fails to resolve the patentability question[s] presented in" this IPR. *Fintiv,* Paper 11 at 15.

The **sixth** factor (other circumstances) also weighs against denial as the merits of Petitioner's arguments are compelling. While the **fifth** factor (parties) may weigh slightly in favor of denial because the parties are the same as in district court, that is true in almost every IPR, and under the "holistic view" of whether integrity of the system and efficiency is best served, institution is favored.

Samsung Elecs. Co. Ltd. v. Dynamics Inc., IPR2020-00505, Paper 11 at 15 (Aug. 12, 2020).

XI. CONCLUSION

Petitioner requests institution of IPR for the challenged claims based on the grounds specified in this petition.

Respectfully submitted,

Dated: June 28, 2024

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)
Counsel for Petitioner

XII. APPENDIX A (CLAIM LISTING)

Claim 10

[10.pre] "A method for retrieving data location information for data stored in a distributed network, comprising the steps of:"

[10.a] "a) receiving at a first client a data query for retrieving data associated with an identification string, wherein the data is stored at a data repository in the distributed network and wherein a location string associated with the identification string of the data is stored in at least one of a plurality of data location servers;"

[10.b] "b) transmitting a data location request from the first client to a first data location server to retrieve the location string associated with the identification string in the data query, the data location request including the identification string;"

[10.c] "c) if the first data location server does not possess the location string, transmitting a redirect message to the first client, the redirect message containing information for use by the first client to calculate a location of a second data location server, wherein the second data location server contains the location string;"

[10.d] "d) calculating the location of the second data location server at the first client; and"

[10.e] "e) transmitting the data query from the first client to the second data location server."

Claim 11

"The method of claim 10, wherein transmitting the redirect message comprises transmitting a data location server table to the first client."

Claim 12

"The method of claim 11, wherein calculating the location of the second data location server comprises calculating the location of the second data location server with a function commonly known to the first data location server and the first client and based on the data location server table and the identification string."

Claim 13

"The method of claim 12, wherein the function comprises a hash function and wherein the first client applies the hash function to the identification string and the data location server table to obtain the location of the second data location server."

Claim 14

"The method of claim 10, wherein transmitting the redirect message comprises transmitting a function to the first client."

Claim 15

"The method of claim 14, wherein calculating at the first client the location of the second data location server comprises applying the transmitted function to the identifier string."

Claim 16

"The method of claim 15, wherein applying the transmitted function generates a URL of the second data location server."

Claim 17

[17.pre] "A system for retrieving data location information for data stored in a distributed network, the system comprising:"

[17.a] "a plurality of data repositories configured to store data, wherein the data is associated with a respective identifier string in each data repository;"

[17.b] "a data location sewer [sic] network having a plurality of data location servers, each of the plurality of data location servers containing location strings associated with respective identifier strings

and each of the plurality of data location servers having computer executable code configured to execute the following steps:"[16]

[17.c] "in response to receiving a data location request from a client to retrieve a location string associated with an identification string provided in the data location request, transmitting a redirect message to the client if the identification string is not associated with a location string at the data location server, wherein the redirect message contains information for use by the client to calculate a location of a different data location server in the plurality of data location servers, wherein the different data location server contains the location string."

Claim 18

[18.pre] "A system for retrieving data location information for data stored in a distributed network, the system comprising:"

[18.a] "a data repository configured to store data, wherein the data is associated with an identifier string;"

[16] Petitioner assumes that "data location sewer network" is intended to refer to a "data location server network."

[18.b] "a client responsive to a data query to query a data location server for location information associated with the identifier string;"

[18.c] "a data location server network comprising a plurality of data location servers, at least one of the plurality of data location servers containing location information associated with the identifier string, wherein each of the plurality of data location servers comprises computer executable code configured to execute the following steps in response to receiving a data location request from the client:"

[18.d] "if the data location server contains the location string associated with the identification string provided in the data location request, the data location server transmits location information for use by the client to calculate a location of the data associated with the identification string;"

[18.e] "if the data location server does not contain the location string associated with the identification string, the location server transmits a redirect message to the client, wherein the redirect message contains redirect information for use by the client to calculate a location of a different data location server in the plurality of data location servers, wherein the different data location server contains the location string."

Claim 19

"The system of claim 18, wherein the client and the plurality of data location servers each comprise a function commonly known to the client and the plurality of data location servers and wherein the client is configured to apply the commonly known function to the location information or redirect information."

Claim 20

"The system of claim 19, wherein the redirect message comprises a data location server table."

Claim 21

"The system of claim 20, wherein the commonly known function comprises a hash function and wherein the client is configured to apply the hash function to the identification string and the data location server table to obtain the location of the different data location server."

Claim 22

"The system of claim 19, wherein the redirect message comprises a transmitted function for use by the client."

Claim 23

"The system of claim 22, wherein the client is configured to calculate the location of the different data location server by applying the transmitted function to the identifier string."

Claim 24

"The system of claim 18, wherein the location information comprises a portion of a hash table distributed over the plurality of data location servers."

CERTIFICATE OF COMPLIANCE

Pursuant to 37 C.F.R. § 42.24(d), the undersigned certifies that the foregoing Petition for *Inter Partes* Review of U.S. Patent No. 7,103,640 contains, as measured by the word-processing system used to prepare this paper, 13,382 words. This word count does not include the items excluded by 37 C.F.R. § 42.24 as not counting towards the word limit.

Respectfully submitted,

Dated: June 28, 2024

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)
Counsel for Petitioner

CERTIFICATE OF SERVICE

I hereby certify that on June 28, 2024, I caused a true and correct copy of the foregoing Petition for *Inter Partes* Review of U.S. Patent No. 7,103,640 and supporting exhibits to be served via express mail on the Patent Owner at the following correspondence address of record as listed on Patent Center:

Reichman Jorgensen Lehman & Feldberg LLP
100 Marine Parkway
Suite 300
Redwood City, CA 94065

A courtesy copy was also sent via electronic mail to the Patent Owner's litigation counsel at the following addresses:

Christine E Lehman (clehman@reichmanjorgensen.com)
Jaime Francisco Cardenas-Navia (jcardenas-navia@reichmanjorgensen.com)
Gina H Cremon (gcremona@reichmanjorgensen.com)
Khue Hoang (khoang@reichmanjorgensen.com)

Reichman Jorgensen Lehman & Feldberg LLP

Respectfully submitted,

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)
Counsel for Petitioner