

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner

v.

SANDPIPER CDN, LLC,
Patent Owner

Case IPR2025-00860
Patent No. 10,924,573

PETITION FOR *INTER PARTES* REVIEW

TABLE OF CONTENTS

I.	PRECISE RELIEF REQUESTED	1
II.	GROUND FOR STANDING.....	2
III.	THE '573 PATENT.....	2
	A. Overview	2
	B. Prosecution History	7
	C. Priority Date	8
IV.	POSITA	12
V.	CLAIM CONSTRUCTION	12
VI.	THE CHALLENGED CLAIMS ARE UNPATENTABLE AND SHOULD BE CANCELLED	13
	A. Ground 1: Claims 1-20 Are Anticipated by Newton-471	13
	1. Overview of Newton-471	13
	2. Independent Claim 1	14
	3. Claim 2: “ <i>The method as recited in claim 1 wherein the popularity designation associated with the requested resource indicates that the requested resource is not popular.</i> ”	23
	4. Claim 3	24
	5. Claim 4: “ <i>The method as recited in claim 1 wherein the second server is a metro mid-tier server of the content delivery network.</i> ”	26
	6. Claim 5: “ <i>The method as recited in claim 1 wherein the second server is a peer server of the first tier of servers of the content delivery network.</i> ”	26

7.	Claim 6: “ <i>The method as recited in claim 1 wherein the popularity designation associated with the requested resource comprises an indication whether a current popularity value for the resource exceeds a first predetermined popularity threshold.</i> ”	27
8.	Claim 7: “ <i>The method as recited in claim 6 wherein the popularity service is associated with the second server of the content delivery network.</i> ”	27
9.	Claim 8: “ <i>The method as recited in claim 6 wherein the popularity service is associated with the first server of the first tier of servers of the content delivery network.</i> ”	28
10.	Claim 9: “ <i>The method as recited in claim 1 wherein the content delivery network further comprises a second tier of servers distinct from the first tier of servers, and wherein the second server is in the second tier of servers.</i> ”	28
11.	Claim 10.....	29
12.	Independent Claim 11	30
13.	Claims 12-18	32
14.	Independent Claim 19	32
15.	Claim 20: “ <i>The method as recited in claim 19 wherein the second server of the content delivery network is a content server storing the requested content.</i> ”	33
B.	Ground 2: Dilley, Pai, and Wang Render Obvious Claims 1-20	34
1.	Overview of Dilley.....	34
2.	Overview of Pai	35
3.	Overview of Wang.....	35
4.	Motivation to Combine	36

5.	Independent Claim 1	41
6.	Claim 2: “ <i>The method as recited in claim 1 wherein the popularity designation associated with the requested resource indicates that the requested resource is not popular.</i> ”	55
7.	Claim 3	59
8.	Claim 4: “ <i>The method as recited in claim 1 wherein the second server is a metro mid-tier server of the content delivery network.</i> ”	64
9.	Claim 5: “ <i>The method as recited in claim 1 wherein the second server is a peer server of the first tier of servers of the content delivery network.</i> ”	67
10.	Claim 6: “ <i>The method as recited in claim 1 wherein the popularity designation associated with the requested resource comprises an indication whether a current popularity value for the resource exceeds a first predetermined popularity threshold.</i> ”	70
11.	Claim 7: “ <i>The method as recited in claim 6 wherein the popularity service is associated with the second server of the content delivery network.</i> ”	72
12.	Claim 8: “ <i>The method as recited in claim 6 wherein the popularity service is associated with the first server of the first tier of servers of the content delivery network.</i> ”	73
13.	Claim 9: “ <i>The method as recited in claim 1 wherein the content delivery network further comprises a second tier of servers distinct from the first tier of servers, and wherein the second server is in the second tier of servers.</i> ”	74
14.	Claim 10	75
15.	Independent Claim 11	77
16.	Claims 12-18	82

17.	Independent Claim 19	83
18.	Claim 20: “ <i>The method as recited in claim 19 wherein the second server of the content delivery network is a content server storing the requested content.</i> ”	85
VII.	THE BOARD SHOULD NOT EXERCISE DISCRETION TO DENY INSTITUTION	86
A.	35 U.S.C. §314(a) Does Not Favor Denial	86
B.	35 U.S.C. §325(d) Does Not Favor Denial	88
VIII.	MANDATORY NOTICES UNDER 37 C.F.R. §42.8.....	89
A.	Real Parties-in-Interest	89
B.	Related Matters.....	89
C.	Lead and Back-Up Counsel, and Service Information	90
IX.	CONCLUSION.....	91

LIST OF EXHIBITS

<u>Exhibit No.</u>	<u>Description</u>
EX-1001	U.S. Patent No. 10,924,573 (“the ’573 patent”)
EX-1002	U.S. Patent Publication No. 2016/0094471 to Newton et al. (“Newton-471”)
EX-1003	U.S. Patent No. 7,133,905 to Dilley (“Dilley”)
EX-1004	U.S. Patent Publication No. 2006/0271972 to Pai (“Pai”)
EX-1005	U.S. Patent Publication No. 2005/0198250 (“Wang”)
EX-1006	Declaration of Dr. Todd Mowry
EX-1007	Prosecution History of the ’573 patent
EX-1008	U.S. Patent No. 10,218,806 to Newton et al.
EX-1009	U.S. Patent No. 9,762,692 to Fullagar et al.
EX-1010	U.S. Patent No. 8,930,538 to Fullagar et al.
EX-1011	U.S. Provisional Patent Application No. 61/042,412 (“the ’412 provisional”)
EX-1012	U.S. Patent No. 8,667,175 to Thomas et al.
EX-1013	U.S. Patent Publication No. 2006/0015574 to Seed et al.
EX-1014	Curriculum Vitae of Dr. Todd Mowry
EX-1015	Order Granting Stipulation Regarding Amended Complaint and Case Schedule, <i>Sandpiper CDN, LLC v. Google LLC</i> , 2:24-cv-03951, Dkt. 54 (C.D. Cal. Jan. 21, 2025)
EX-1016	Sandpiper’s Disclosures of Asserted Claims & Infringement Contentions dated Jan. 14, 2025 in <i>Sandpiper CDN, LLC v. Google LLC</i> , 2:24-cv-03951.

<u>Exhibit No.</u>	<u>Description</u>
EX-1017	Exhibit D to Sandpiper’s Disclosures of Asserted Claims & Infringement Contentions dated Jan. 14, 2025 in <i>Sandpiper CDN, LLC v. Google LLC</i> , 2:24-cv-03951.
EX-1018	Letter to Robert H. Reckers <i>re Sandpiper CDN LLC v. Google LLC</i> , No. 2:24-3951-AB-RAO (C.D. Cal.) (Apr. 29, 2025)
EX-1019	Duane Wessels, Web Caching, O’REILLY (Torkington et al. 1st ed. 2001)
EX-1020	Lv, Z., Yang, et al., Multimedia Delivery Grid: A Novel Multimedia Delivery Scheme, In: Li, M., Sun, XH., Deng, Qn., Ni, J. (eds) GRID AND COOPERATIVE COMPUTING. GCC 2003. LECTURE NOTES IN COMPUTER SCIENCE, vol 3032. Springer, Berlin, Heidelberg (2004), https://doi.org/10.1007/978-3-540-24679-4_46
EX-1021	Prosecution History of U.S. Patent Application No. 12/408,681
EX-1022	U.S. Patent Publication No. 2007/0055764 A1

LISTING OF CHALLENGED CLAIMS

Claim 1	
1(preamble)	A method of content delivery in a content delivery network comprising:
1(a)	receiving, at a first server of a first tier of servers of the content delivery network, a request from a requesting device for a resource available from the content delivery network;
1(b)	accessing a popularity service associated with the content delivery network to determine a popularity designation associated with the requested resource;
1(c)	requesting the resource from a second server of the content delivery network;
1(d)	processing, at the first server of the first tier of servers, a redirect instruction from the second server of the content delivery network to obtain the resource from a content server of the content delivery network;
1(e)	receiving an instruction to not cache the portion of the resource at the first server of the first tier of servers when the portion of the resource is obtained from the content server of the content delivery network; and
1(f)	providing the obtained resource to the requesting device.
Claim 2	
2	The method as recited in claim 1 wherein the popularity designation associated with the requested resource indicates that the requested resource is not popular.
Claim 3	
3(a)	The method as recited in claim 1 wherein the popularity designation associated with the requested resource indicates that the requested resource is popular,

3(b)	the method further comprising: receiving an instruction to cache the resource at the first server of the first tier of servers when the resource is obtained from the content server of the content delivery network.
Claim 4	
4	The method as recited in claim 1 wherein the second server is a metro mid-tier server of the content delivery network.
Claim 5	
5	The method as recited in claim 1 wherein the second server is a peer server of the first tier of servers of the content delivery network.
Claim 6	
6	The method as recited in claim 1 wherein the popularity designation associated with the requested resource comprises an indication whether a current popularity value for the resource exceeds a first predetermined popularity threshold.
Claim 7	
7	The method as recited in claim 6 wherein the popularity service is associated with the second server of the content delivery network.
Claim 8	
8	The method as recited in claim 6 wherein the popularity service is associated with the first server of the first tier of servers of the content delivery network.
Claim 9	
9	The method as recited in claim 1 wherein the content delivery network further comprises a second tier of servers distinct from the first tier of servers, and wherein the second server is in the second tier of servers.

Claim 10	
10(preamble)	The method as recited in claim 3 further comprising:
10(a)	receiving, at the first server of the first tier of servers of the content delivery network, an additional request for the resource available from the content delivery network;
10(b)	determining that the resource is cached at the first server of the first tier of servers; and
10(c)	providing the cached resource from the first server of the first tier of servers in response to the additional request for the resource.
Claim 11	
11(preamble)	A content delivery network comprising:
11(a)	a first tier of servers comprising a first plurality of servers, an edge server of the first tier of servers receiving a request from a requesting device in communication with the first server for a resource available from the content delivery network;
11(b(i))	a second tier of servers comprising a second plurality of servers,
11(b(ii))	a first server of the second plurality of servers receiving a request from the edge server for the content and, in response, transmitting a redirect instruction intended for the requesting device to the edge server to obtain the resource from a content server and transmitting an instruction to not cache the portion of the resource at the edge server when the portion of the resource is obtained from the content server; and
11(c)	a popularity service tracking a popularity designation associated with the requested resource;

11(d)	wherein the edge server processes the redirect instruction to obtain the content from the content server and provides the obtained content to the requesting device.
Claim 12	
12	The content delivery network as recited in claim 11 wherein the popularity designation associated with the requested resource indicates that the requested resource is not popular.
Claim 13	
13(a)	The content delivery network as recited in claim 11 wherein the popularity designation associated with the requested resource indicates that the requested resource is popular and
13(b)	the first server of the second plurality of servers transmits an instruction to cache the resource at the content server of the first tier of servers when the resource is obtained from the content server.
Claim 14	
14	The content delivery network as recited in claim 11 wherein the first server of the second plurality of servers is a metro mid-tier server of the content delivery network.
Claim 15	
15	The content delivery network as recited in claim 11 wherein the first server of the second plurality of servers is a peer server of the first tier of servers of the content delivery network.
Claim 16	
16	The content delivery network as recited in claim 11 wherein the popularity designation associated with the requested resource comprises an indication whether a current popularity value for the resource exceeds a first predetermined popularity threshold.
Claim 17	

17	The content delivery network as recited in claim 16 wherein the popularity service is associated with the first server of the second plurality of servers of the content delivery network.
Claim 18	
18	The content delivery network as recited in claim 16 wherein the popularity service is associated with the content server of the first tier of servers of the content delivery network.
Claim 19	
19(preamble)	A method of content delivery in a content delivery network comprising:
19(a)	receiving, at a first server of a first tier of servers of the content delivery network, a request from a requesting device for a resource available from the content delivery network;
19(b)	accessing a popularity service associated with the first server of the first tier of the content delivery network to determine a popularity designation associated with the requested resource;
19(c)	receiving an instruction to not cache the resource at the first server of the first tier of servers when the portion of the resource is obtained from the content server of the content delivery network;
19(d)	requesting the resource from a second server of the content delivery network based at least on the popularity designation associated with the requested resource; and
19(e)	providing the obtained resource to the requesting device.
Claim 20	
20	The method as recited in claim 19 wherein the second server of the content delivery network is a content server storing the requested content.

I. PRECISE RELIEF REQUESTED

Google LLC (“Petitioner”) requests *inter partes* review under 35 U.S.C. §311 and cancellation of claims 1-20 of U.S. Patent No. 10,924,573 (the “’573 patent”) in view of the following:

<u>Prior Art</u>	
Ref. 1:	<i>Newton-471</i> (EX-1002), U.S. Pub. 2016/0094471, published March 31, 2016, filed September 30, 2015.
Ref. 2:	<i>Dilley</i> (EX-1003), U.S. Patent 7,133,905, published November 7, 2006, filed April 9, 2002.
Ref. 3:	<i>Pai</i> (EX-1004), U.S. Pub. 2006/0271972, published November 30, 2006, filed May 31, 2005.
Ref. 4:	<i>Wang</i> (EX-1005), U.S. Pub. 2005/0198250, published September 8, 2005, filed March 8, 2005.

Each reference is prior art under 35 U.S.C. §102(a)(1) under the proper post-AIA priority date of February 25, 2019, *infra* §III.C (Priority Date). Dilley, Pai, and Wang are also prior art under pre-AIA 35 U.S.C. §102(b) under the ’573 patent’s claimed April 4, 2008, priority date.

<u>Grounds of Unpatentability</u>	
1	Claims 1-20 are anticipated by Newton-471
2	Claims 1-20 would have been obvious over Dilley in view of Pai and Wang

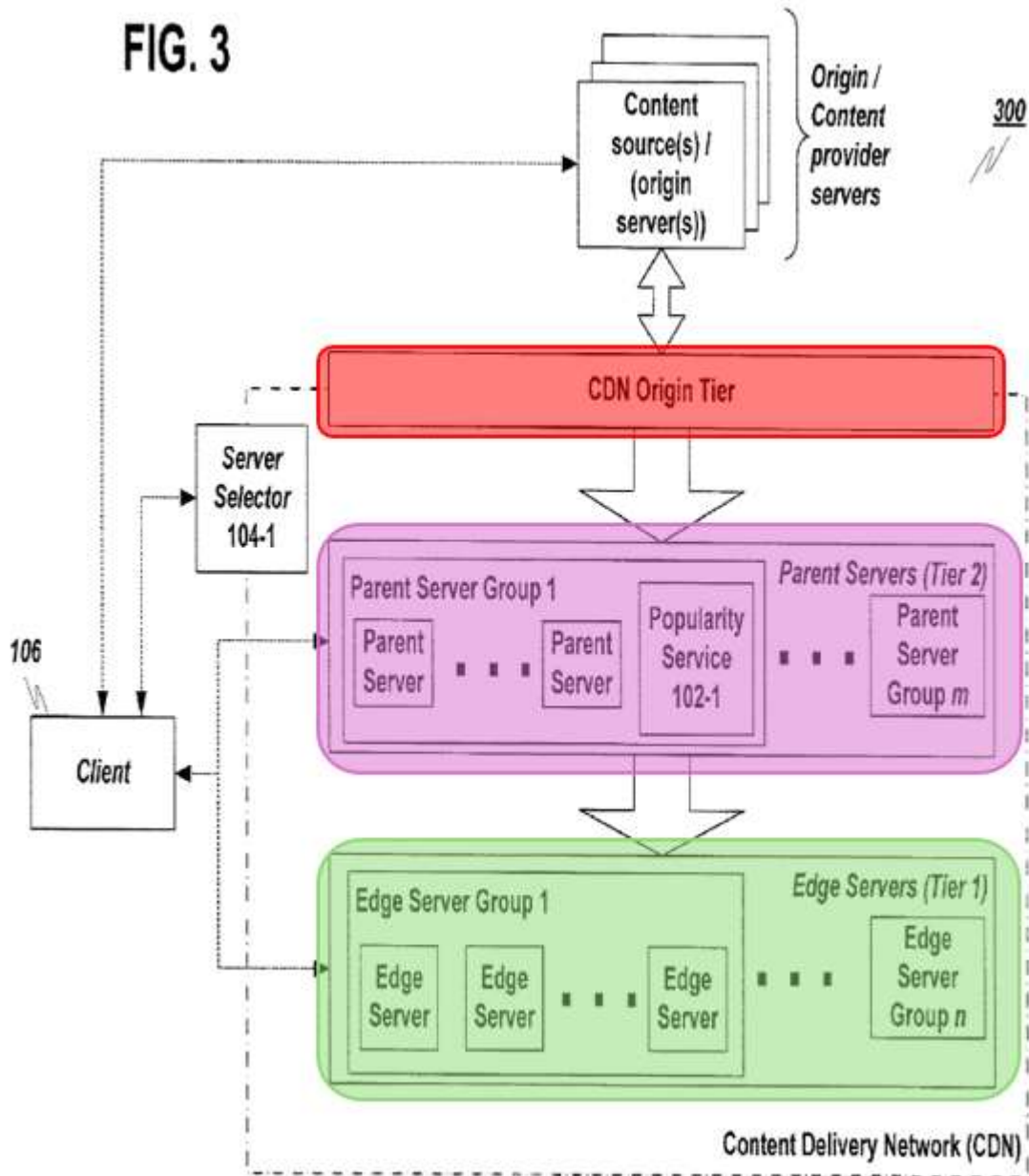
II. GROUNDS FOR STANDING

Petitioner is not barred or estopped from requesting IPR on these grounds.

III. THE '573 PATENT

A. Overview

The '573 patent is generally directed to a content delivery network ("CDN") that includes a popularity service. EX-1001, Abstract. A CDN is a network of servers distributed geographically to deliver web content to users. EX-1001, 2:28-36; EX-1012, 1:13-19. The '573 patent describes a CDN organized into hierarchical tiers, where tier 1 servers represent edge servers, sitting closest to the clients (end-users), the origin tier represents those servers sitting furthest from the clients, and a parent tier could sit in between. EX-1001, 2:60-65, 3:53-57, 4:19-31; EX-1013, ¶[0010] ("edge servers are typically closer (by various measures of distance) to end users").



EX-1001, Fig. 3 (annotated showing **edge servers**, **parent servers**, and **CDN origin tier**); see also *id.*, Fig. 1.

The '573 patent also describes a server selector system that directs user requests to an appropriate edge server based on factors like proximity to the user and server load. EX-1001, 4:47-55.

The edge servers receive incoming requests for content (e.g., web pages or videos) based on the server selector system and attempt to handle these requests directly from their cache¹ when possible. EX-1001, 5:59-6:17. Thus, when a client requests a resource from the system, the first server (e.g., a selected edge server) will receive the request and check whether the content is available from the first server or a nearby peer server (e.g., another edge server in the same group). *Id.* If the resource is available, the content is served to the client from the first server. *Id.*

¹ A cache stores or manages a copy of content stored on origin servers so that future requests for that content may be served faster. EX-1006, ¶43; EX-1019, 15.

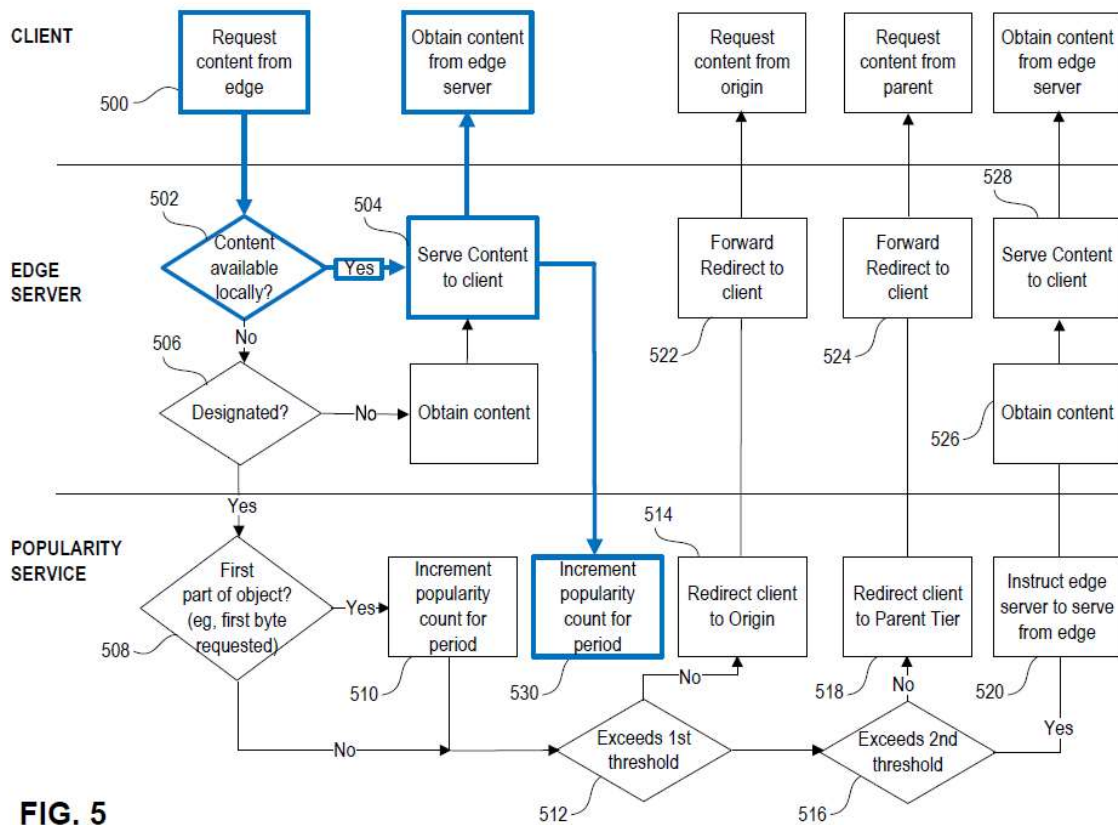


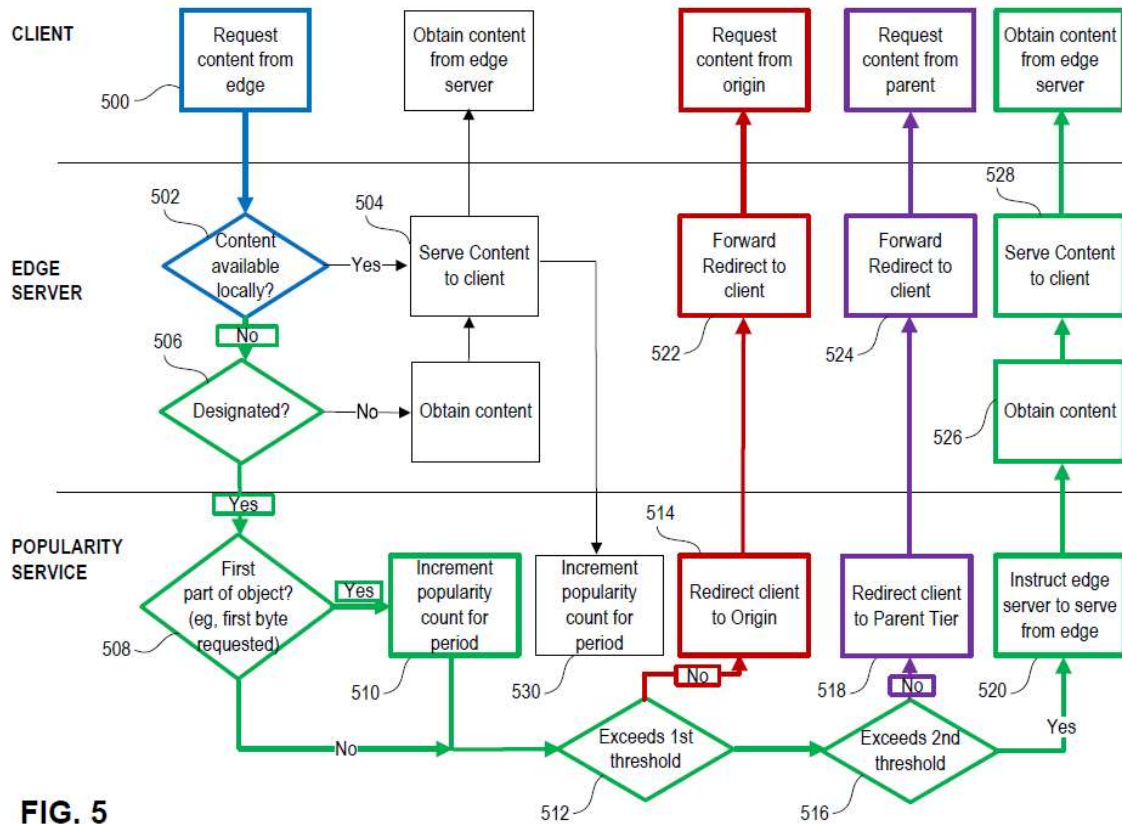
FIG. 5

Id., Fig. 5 (sharpened and annotated).

When the resource is unavailable at this first level—for instance for less frequently accessed content—the ’573 patent will evaluate the content’s popularity using a “popularity service,” and will provide one of three exemplary responses (*id.*, 6:27-34):

First, for unpopular resources (e.g., below a minimum popularity threshold), the request is redirected to an origin server/tier; second, when a resource is somewhat popular (e.g., between minimum and higher popularity thresholds), the client’s request is redirected to a parent server; and third, when a resource is popular

(e.g., exceeding higher popularity threshold), the first server will retrieve the content and serve the request itself. EX-1001, 6:53-7:30.



Id., Fig. 5 (annotating in green the process for popular content being served from edge, annotating in purple the process for mid-level popular content being served from a parent server, and annotating in red the least popular content being served from the origin server/tier). EX-1006, ¶¶41-46.

B. Prosecution History

Application No. 16/284,791 (the “’791 application”), which issued as the ’573 patent, was filed on February 25, 2019 (EX-1001), and examined under pre-AIA provisions. EX-1007, 17.

In a first action, Examiner issued a nonstatutory double patenting rejection and rejected various claims as obvious over prior art not at issue here. EX-1007, 59-78. Examiner did not, however, issue an obviousness rejection for then-pending claim 2, which recited “receiving *an instruction to not cache* the portion of the resource at the first server ... when the portion of the resource is obtained from the content server....” EX-1007, 276 (emphasis added).

In response, Applicant filed a terminal disclaimer to overcome the double patenting rejections and amended the independent claims to incorporate claim 2’s unrejected “instruction to not cache” limitation. EX-1007, 46-47.

Examiner then issued a Notice of Allowance stating that the “prior art of record fails to teach or suggest ‘receiving an *instruction to not cache* the resource at the first server.’” EX-1007, 17 (emphasis added). Specifically, Examiner differentiated “[t]he closest prior art” explaining that while that art “teaches a policy for managing caching content at the edge,” a determination of “whether [to] cache a content item is different from ‘sending an instruction to not cache the content item’ as recited in the claims.” EX-1007, 17. Examiner similarly differentiated

“[t]he other closest prior art” because, although it caches content “based on the popularity value,” accessing “the popularity value is different from ‘sending an instruction to not cache the content item’ as recited in the claims.” EX-1007, 17. Thus, the Challenged Claims were allowed in light of this “instruction to not cache” limitation. EX-1006, ¶¶47-53.

But as discussed below, Newton-471, as well as the other prior art not applied by Examiner, discloses this limitation.

C. Priority Date

The priority date for the challenged claims is February 25, 2019, the filing date of the ’791 application, because new matter—specifically, the “instruction to not cache” limitation—was introduced on that date and has no support in any application to which the ’573 patent claims priority. EX-1006, ¶¶54-56.

The ’791 application claims priority as a continuation of an application that issued as U.S. Patent 10,218,806 (the “’806 patent”), which claims priority as a continuation of an application that issued as U.S. Patent 9,762,692 (the “’692 patent”), which claims priority as a continuation-in-part of an application that issued as U.S. Patent 8,930,538 (the “’538 patent”), which ultimately claims priority to U.S. Provisional Application 61/042,412 (the “’412 provisional”), filed April 4, 2008. EX-1006, ¶¶48, 56.

A patent's claims are not entitled to an earlier priority date merely because a patent owner claims priority. *See Dynamic Drinkware, LLC v. National Graphics, Inc.*, 800 F.3d 1375, 1379-1380 (Fed. Cir. 2015). Here, Sandpiper must show the earlier application provides written description support under 35 U.S.C. §112(a). 35 U.S.C. §120; *Dynamic Drinkware*, 800 F.3d at 1379–80; *Paice LLC v. Ford Motor Co.*, 881 F.3d 894, 906 (Fed. Cir. 2018); *PowerOasis, Inc. v. T-Mobile USA, Inc.*, 522 F.3d 1299, 1306–11 (Fed. Cir. 2008).

The test for written description sufficiency is whether the prior application's disclosure "reasonably conveys to those skilled in the art that the inventor had possession of the claimed subject matter as of the filing date." *Ariad Pharms., Inc. v. Eli Lilly & Co.*, 598 F.3d 1336, 1351 (Fed. Cir. 2010) (en banc); *Augustine Medical, Inc. v. Gaymar Indus., Inc.*, 181 F.3d 1291, 1302–03 (Fed. Cir. 1999).

Sandpiper cannot meet this burden. The written description of the '573 patent has no support for an "instruction to not cache"; the limitation appears only in the claims. *See generally* EX-1001; EX-1006, ¶56. The '573 patent's immediate parent applications, which resulted in the '806 and '692 patents, also fail to disclose an "instruction to not cache," as each shares a specification with the '573 patent. *Compare* EX-1008 *with* EX-1009.

Although new matter was added in the application that resulted in the '692 patent (specifically a description of "processing steps associated with a multi-tiered

content-delivery network,” EX-1009, 2:3-4, 11:55-12:28, Fig. 7), none of the new matter discusses or provides support for the instruction to not cache limitation. EX-1006, ¶¶57-58.

The ’538 patent’s disclosure is substantively the same as the ’412 provisional application that is the earliest claimed priority application. But neither the ’412 provisional nor, by extension, the intervening three applications to which the ’573 patent claims priority support the “instruction not to cache” limitation found in each challenged claim, either directly or by dependency. EX-1006, ¶59.

The ’412 provisional discusses a determination of whether to cache content at the first server and provides examples for when it will “obtain and cache[] the resource.” EX-1011, ¶¶[0012], [0044]. It similarly contemplates a “rule base” to identify classes of content (e.g., content that will quickly stale) that will be marked as “unpopular,” regardless of actual popularity, to prevent caching. EX-1011, ¶[0062] (“a subscriber may not want to pay for certain content to be served from the edge, regardless of its popularity, and may set a rule accordingly” (i.e., “setting the thresholds for that particular content to prevent it from ever being cached at the edge”)); *see also id.* ¶[0052] (providing an example of “HTTP POST requests”); EX-1006, ¶¶60-64; EX-1019, 43.

But the specification is silent as to sending or receiving instructions to not cache a resource. *Boston Scientific Corp. v. Johnson & Johnson*, 647 F.3d 1353,

1366 (Fed. Cir. 2011) (concluding that one must make an “‘objective inquiry into the four corners of the specification’” from the perspective of a POSITA to determine written description support). The Examiner pointed out that determining “whether [to] cache a content item”—e.g, based on configured policy rules or a popularity value, as in the ’412 provisional (and related applications)—“is *different* from ‘sending an instruction to not cache the content item’ as recited in the claims.” EX-1007, 17 (emphasis added).

Thus, none of the priority documents provide written description support for the “instruction to not cache” limitations in every challenged claim. EX-1006, ¶65; *Dr. Reddy’s Labs. S.A. v. Indivior UK Ltd.*, IPR2019-00329, Paper 49, 82 (P.T.A.B. June 2, 2020) (earliest possible priority was filing date of last patent application where no application in priority chain provided written description support). Each of the challenged claims is therefore entitled to a priority date no earlier than the ’573 patent’s filing—February 25, 2019.

Accordingly, under the actual priority date of February 25, 2019, all of the references are prior art under 35 U.S.C. § 102(a)(1). But even if the ’573 patent were entitled to its claimed priority date of April 4, 2008, the Ground 2 references remain prior art under pre-AIA 35 U.S.C. § 102(b).

IV. POSITA

A person of ordinary skill in the art (“POSITA”) at the time of the alleged invention (including either April 4, 2008, or February 25, 2019), would have had at least a bachelor’s degree in computer science, electrical engineering, or a related field, and at least two years of work or research experience in the field of content delivery management or networks. Work experience can substitute for formal education and additional formal education can substitute for work experience. EX-1006, ¶¶70.

V. CLAIM CONSTRUCTION

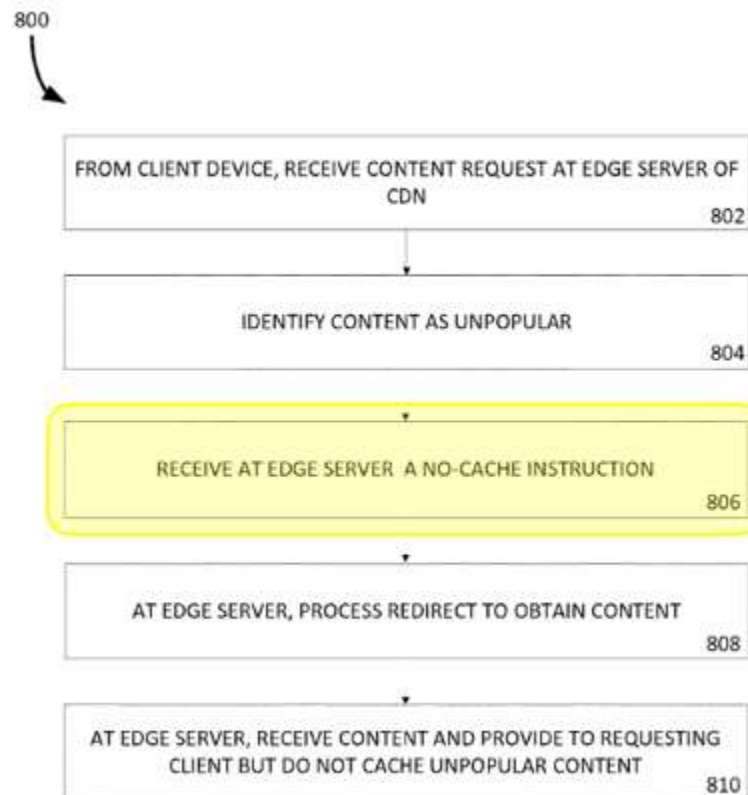
The Board construes claims under *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc), under which terms receive their plain and ordinary meaning as understood by one of ordinary skill in the art, consistent with the disclosure and prosecution history. *Id.* at 1314-19; *see* 37 C.F.R. §42.100(b) (2018). Claims should only be construed to the extent necessary to resolve a controversy. *Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017). No claim terms need to be construed by the Board at this time, and all should be given their ordinary meanings for purposes of this proceeding. EX-1006, ¶¶67-69.

VI. THE CHALLENGED CLAIMS ARE UNPATENTABLE AND SHOULD BE CANCELLED

A. Ground 1: Claims 1-20 Are Anticipated by Newton-471

1. Overview of Newton-471

Newton-471 is a published application that shares largely the same written description and figures with the '573 patent, while including additional support related to the “instruction to not cache” limitations. *Compare* EX-1002, *with* EX-1001. Specifically, Newton-471’s Figure 8 and its corresponding description regarding an “instruction not to cache” is absent from the '573 patent and its priority applications:



EX-1002, Fig. 8 (annotated).

Figure 8 shows the edge server receiving a specific no-cache instruction (step 806) and providing the requested content to the client without caching it at the edge (step 810). EX-1002, Fig. 8, ¶¶[0098]-[0099] (explaining “mid-tier server 708 may also return a cache or no-cache instruction” for unpopular content). The ’573 patent and its prior applications fail to include this (or any other description) regarding a no-cache instruction. And because the ’573 patent cannot claim priority before February 25, 2019, Newton-471, which published on March 31, 2016, is §102(a)(1) prior art. EX-1006, ¶¶74-77.

2. Independent Claim 1

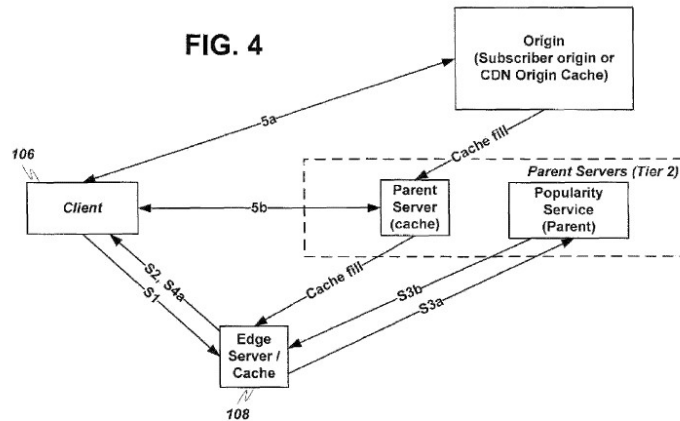
a. *1(preamble): A method of content delivery in a content delivery network comprising:*

Newton-471, similar to the ’573 patent, discloses management of content delivery in a CDN. EX-1002, ¶¶[0010]-[0012] (describing “content delivery network” and “method for content delivery”), claims 1, 11, 19 (same); EX-1006, ¶¶78-80.

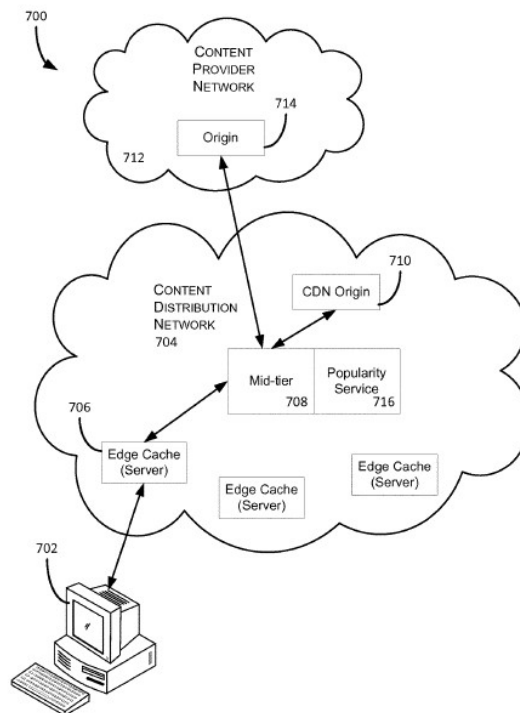
b. *1(a): receiving, at a first server of a first tier of servers of the content delivery network, a request from a requesting device for a resource available from the content delivery network;*

Newton-471 explains its “method includes the operations of receiving, at a first server of a first tier of servers of the content delivery network, a request from a requesting device for a resource available from the content delivery network.”

EX-1002, ¶¶[0010]-[0012], [0048], [0096], [0100] (providing edge servers receive client requests); *see also id.*, claims 1, 11, 19.

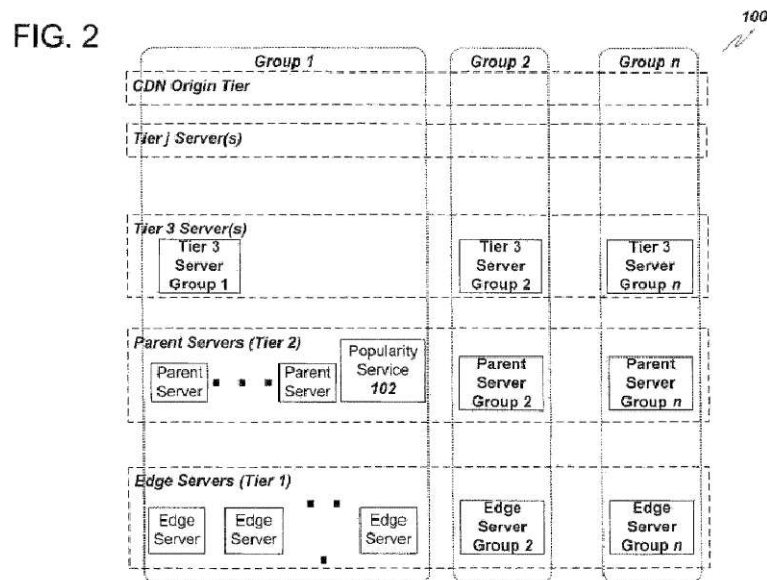
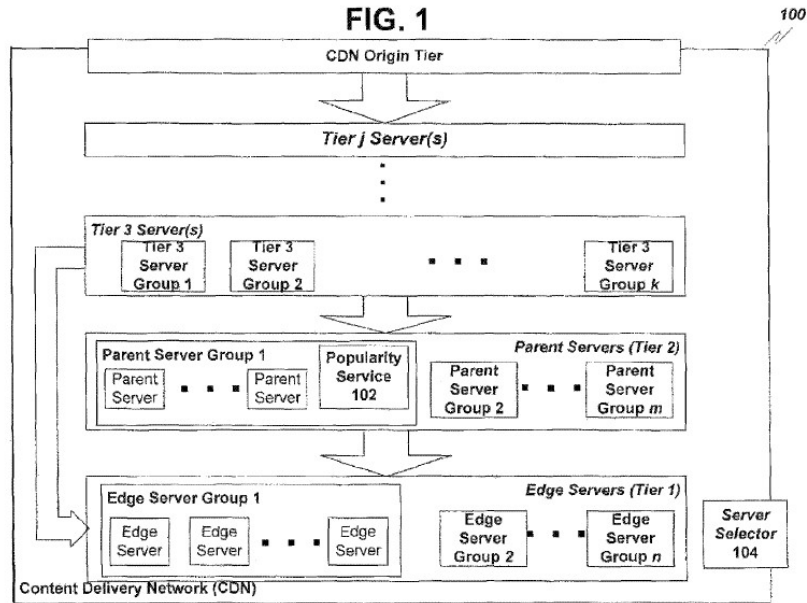


EX-1002, Fig. 4.



EX-1002, Fig. 7 (depicting edge server receiving client request).

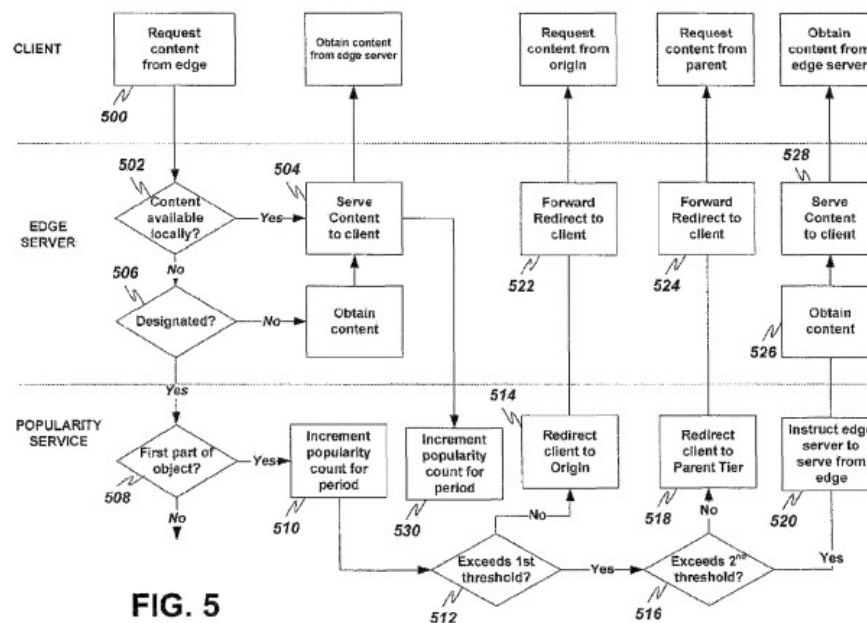
Newton-471's hierarchical CDN "may have one or more tiers of servers," including, "Tier 1 [through] Tier j," where each tier "may comprise a number of servers organized into server groups," such that the first tier of servers can include a plurality of edge servers. EX-1002, ¶¶[0029]-[0030].



EX-1002, Figs. 1-2; EX-1006, ¶¶81-84.

- c. ***1(b): accessing a popularity service associated with the content delivery network to determine a popularity designation associated with the requested resource;***

Newton-471 explains that its “method includes the operations” of “accessing a popularity service associated with the content delivery network to determine a popularity designation associated with the requested resource.” EX-1002, ¶¶[0010]-[0012]. Newton-471’s Figure 5 and supporting description is identical to Figure 5 in the ’573 patent and details the claimed popularity service:



EX-1002, Fig. 5.

Newton-471 explains “the request is sent to the popularity service 102” when a requested object is not available at the edge. EX-1002, ¶¶[0050], [0096]-[0098], [0100]. The popularity service determines the current popularity value. EX-1002,

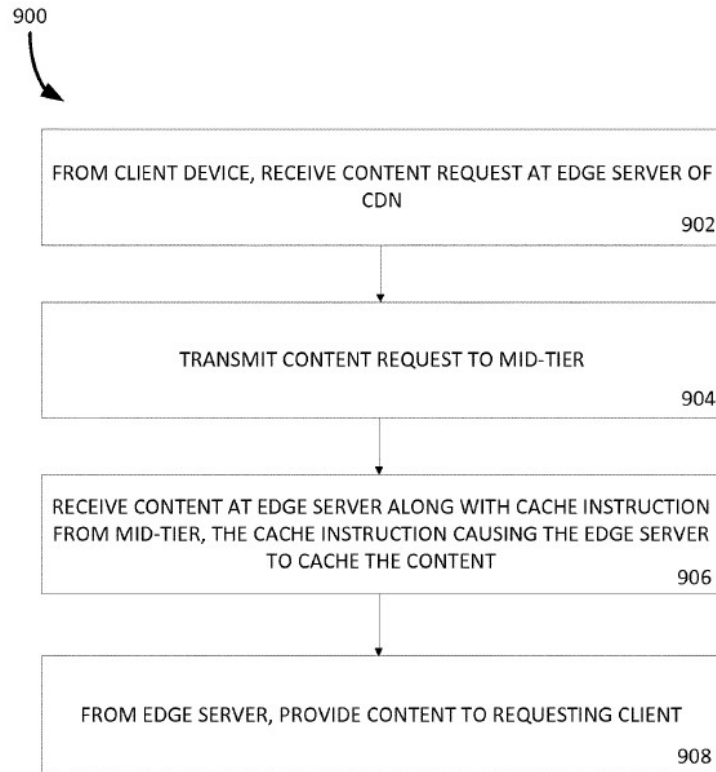
¶¶[0053]-[0057] (three possible popularity responses); *id.*, claims 1, 11, 19; EX-1006, ¶¶85-88.

d. *1(c): requesting the resource from a second server of the content delivery network;*

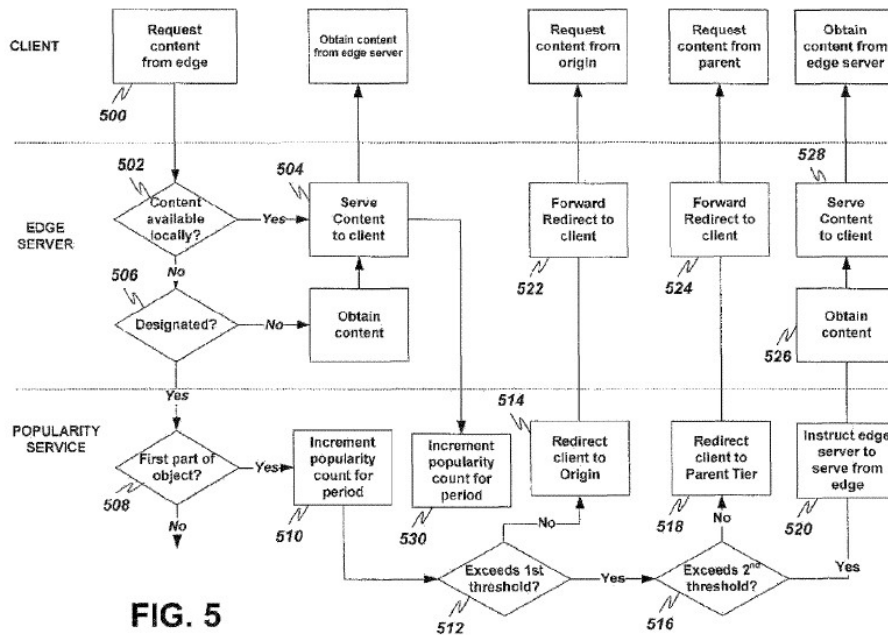
Newton-471 explains that its “method includes the operations” of “requesting the resource from a second server of the content delivery network.” EX-1002, ¶¶[0010]-[0012]; *see also id.*, claims 1, 11, 19.

It does this in at least two ways. EX-1006, ¶¶89-90. First, Newton-471 explains that, after receiving a request, “[t]he edge server 108 checks to see if the object is present (locally or on a peer).” EX-1002, ¶[0048]. In this instance, the peer server would be a second server of the CDN.

Second, Newton-471 explains that if the content is not available at the first server, the request may be “redirect[ed] to a parent or origin server,” as illustrated in Figures 9 and 5. EX-1002, ¶[0061]; *see also* EX-1002, ¶¶[0055]-[0057], [0096]; EX-1006, ¶¶90-91.



EX-1002, Fig. 9.



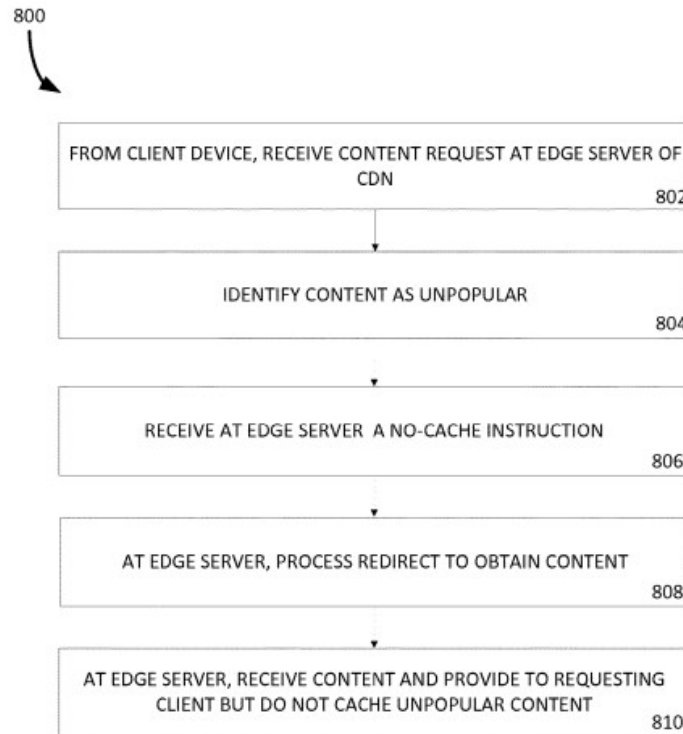
EX-1002, Fig. 5 (flowchart depicting content requested from origin or parent server based on popularity).

- e. ***1(d): processing, at the first server of the first tier of servers, a redirect instruction from the second server of the content delivery network to obtain the resource from a content server of the content delivery network;***

To the extent the '573 patent contains written description support for this limitation², it is disclosed by Newton-471, which explains its “method includes” “processing, at the first server of the first tier of servers, a redirect command from the second server of the content delivery network to obtain the resource from a content server of the content delivery network.”³ EX-1002, ¶[0010]; *see also id.*, claims 1, 11; EX-1006, ¶¶92-95. Newton-471 discloses that “[t]he edge server 706 may process the redirect [instruction],” received from the second server, as shown in operation 808. EX-1002, ¶¶[0097]-[0099]; *see also id.*, ¶¶[0095]-[0096], [0101].

² Petitioner does not concede, in either ground, that this claim meets that the statutory requirements of 35 U.S.C. §112.

³ While Newton-471’s Summary discloses “a redirect command,” it later claims “a redirect instruction.” EX-1002, claim 1 (“processing . . . a redirect instruction from the second server”).



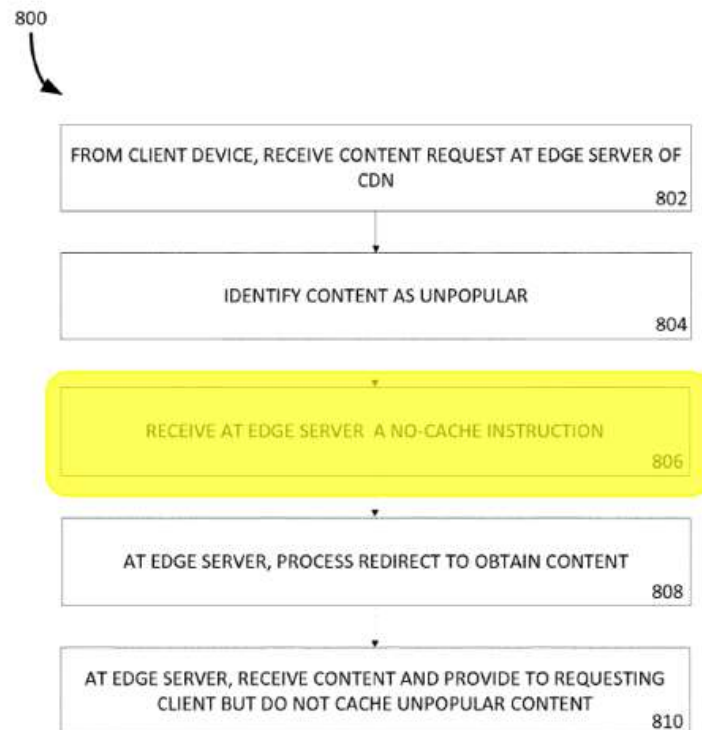
EX-1002, Fig. 8.

Newton-471 further explains, depending on popularity, “the popularity service sends the edge server an instruction (e.g., HTTP 302) to redirect the client’s request to a parent server.” EX-1002, ¶¶[0055]-[0057], [0061].

- f. ***1(e): receiving an instruction to not cache the portion of the resource at the first server of the first tier of servers when the portion of the resource is obtained from the content server of the content delivery network; and***

While this limitation does not have support in the ’573 patent’s priority documents, it is disclosed in Newton-471. Newton-471’s Figure 8 depicts a method of serving content from the CDN “where the content is not cached.” EX-1002,

¶¶[0095]-[0096]; *see also id.*, ¶[0012]; EX-1006, ¶¶96-99. In step 806, the edge server will receive a “no-cache instruction” and will accordingly serve, but not cache, the requested content. EX-1002, ¶¶[0098]-[0099].

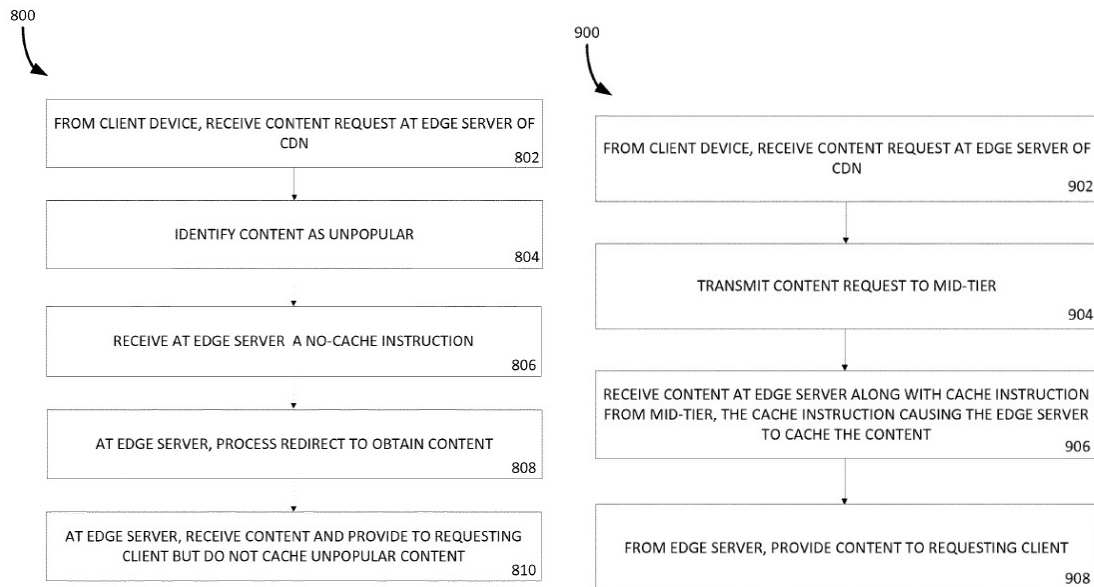


EX-1002, Fig. 8 (annotated).

Newton-471 discloses that when an edge server does not have the requested content, it contacts another server (such as a mid-tier server) to receive it. EX-1002, ¶[0098]. “In addition to returning the requested content, the mid-tier server 708 may also return a cache *or no-cache instruction*.” *Id.* (emphasis added) Specifically, it states “[f]or unpopular content, the mid-tier 708 returns a no-cache instruction,” and “edge server 706 does not cache the content in response to the no cache instruction.” EX-1002, ¶[0099]; *see also id.*, claims 2, 19.

g. 1(f): providing the obtained resource to the requesting device.

Newton-471 explains its “method includes” “providing the obtained resource to the requesting device.” EX-1002, ¶¶[0010]-[0012]; *see id.*, claim 1. Newton-471’s system intends to provide the requested content to the client (*i.e.*, requesting device). EX-1002, ¶¶[0047]-[0048], [0052], [0058], [0099]-[0101]. Figures 8 and 9 conclude with providing content to the requesting client:



EX-1002, Figs. 8 (left), 9 (right); EX-1006, ¶¶100-102.

3. Claim 2: “The method as recited in claim 1 wherein the popularity designation associated with the requested resource indicates that the requested resource is not popular.”

Newton-471 explains that the popularity designation associated with the requested resources indicates the resource’s popularity, including whether that resource is or is not popular. EX-1002, ¶¶[0096]-[0097] (explaining the popularity

service determines if requested resource is unpopular); *id.*, ¶¶[0054]-[0057], Fig. 8 (step 804), claims 2, 12; EX-1006, ¶¶103-105.

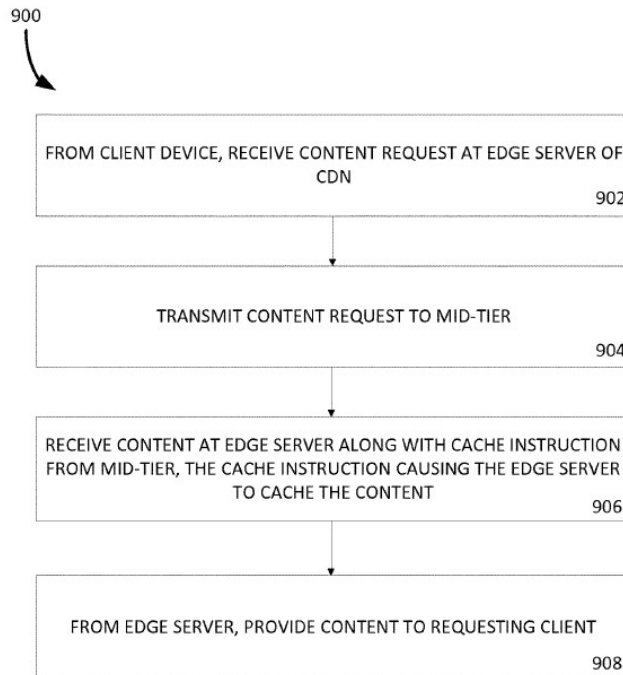
4. Claim 3

- a. ***3(a): The method as recited in claim 1 wherein the popularity designation associated with the requested resource indicates that the requested resource is popular,***

Newton-471 explains that the popularity designation associated with the requested resources indicates the resource's popularity, including whether that resource is popular. EX-1002, ¶[0100] ("determin[ing] that the content is popular based on the popularity service"); *see also id.*, ¶¶[0054]-[0057] (determining content is unpopular, popular, or somewhere in-between), claims 3, 13; EX-1006, ¶¶106-108.

- b. ***3(b): the method further comprising: receiving an instruction to cache the resource at the first server of the first tier of servers when the resource is obtained from the content server of the content delivery network.***

Newton-471 explains that “[i]f the content is or has become popular as determined by the popularity service 716, then the content is returned to the edge server 706 with a cache instruction.” EX-1002, ¶[0101]; *id.*, ¶[0092], claims 3, 13.⁴



EX-1002, Fig. 9 (step 906).

⁴ Based on its dependency, this claim creates an inconsistency, requiring an instruction *to cache* and *not to cache* the resource. For purposes of this Petition, and without conceding the statutory requirements of 35 U.S.C. §112 are met for claims 3 or 10, Petitioner addresses this limitation assuming dependent claim 3 can be reconciled with claim 1. EX-1006, ¶111.

Indeed, the instruction to cache is received when the resource is obtained from the content server because Newton-471 explains, “[r]egardless of from which component of the CDN 704 the content is received, the edge server 706 receives the content in operation 906 and, in response to the cache instruction, caches the popular content at the content server.” EX-1002, ¶[0101]; EX-1006, ¶¶109-112.

5. Claim 4: “*The method as recited in claim 1 wherein the second server is a metro mid-tier server of the content delivery network.*”

Newton-471 discloses this limitation. *See e.g.*, EX-1002, ¶¶[0093]-[0094] (explaining mid-tier server can be a “metro mid-tier server 708” and providing the benefits of using such a “metro mid-tier server”), claims 4, 14; EX-1006, ¶¶113-114.

6. Claim 5: “*The method as recited in claim 1 wherein the second server is a peer server of the first tier of servers of the content delivery network.*”

Newton-471 discloses this limitation for at least the same reasons provided above for limitation 1(c), when the second server is a peer server of the first tier of servers. *Supra* §VI.A.2.d; *see also* EX-1002, claims 5, 15; EX-1006, ¶¶115-116.

7. **Claim 6: “*The method as recited in claim 1 wherein the popularity designation associated with the requested resource comprises an indication whether a current popularity value for the resource exceeds a first predetermined popularity threshold.*”**

Newton-471 discloses this limitation for at least the same reasons provided above for limitation 1(b), claim 2, and limitation 3(a). *Supra* §§VI.A.2.c, VI.A.3, VI.A.4.a; *see also* EX-1002, Fig. 5, claims 6, 16. Indeed, Newton-471 provides three potential outcomes in response to a popularity determination: the object is (1) unpopular, falling below a minimal threshold; (2) mid-tier popular, exceeding the minimal threshold but not a second, mid-tier threshold; or (3) popular, exceeding the mid-tier threshold. EX-1002, ¶¶[0054]-[0057], [0063]; EX-1006, ¶¶117-118.

8. **Claim 7: “*The method as recited in claim 6 wherein the popularity service is associated with the second server of the content delivery network.*”**

Newton-471 explains that the popularity service can be “associated with one or more of the server groups in one or more tiers.” EX-1002, ¶[0039]. It provides an “exemplary embodiment” wherein “some of the parent server groups have popularity services 102 associated therewith.” EX-1002, ¶[0039]. Thus, Newton-471’s popularity service is associated with the second server, when the second server is a parent server, and it further contemplates that “the popularity service 102 may be integrated into one or more of the servers in the group, [and i]n general, the popularity service 102 may be associated or integrated into any server or server

group.” EX-1002, ¶¶[0039], [0064], [0077], Figs. 1-4, claims 7, 17; EX-1006, ¶¶119-121.

9. Claim 8: “*The method as recited in claim 6 wherein the popularity service is associated with the first server of the first tier of servers of the content delivery network.*”

Newton-471 discloses this limitation for at least the same reasons as above for claim 7. *Supra* §VI.A.8. Newton-471 suggests that the popularity service may be associated with any server. EX-1002, ¶[0039], claims 8, 18; EX-1006, ¶¶122-123.

10. Claim 9: “*The method as recited in claim 1 wherein the content delivery network further comprises a second tier of servers distinct from the first tier of servers, and wherein the second server is in the second tier of servers.*”

Newton-471 discloses this limitation for at least the same reasons as above for limitations 1(a) and 1(c), wherein the second server is a parent server. *Supra* §§VI.A.2.b, VI.A.2.d; *see also* EX-1002, claims 9, 11. Newton-471 describes a hierarchical CDN with multiple tiers of servers, including “Tier 1 [through] Tier j.” EX-1002, ¶[0029]-[0030]; EX-1006, ¶¶124-125.

11. Claim 10

- a. ***10(a): The method as recited in claim 3 further comprising: receiving, at the first server of the first tier of servers of the content delivery network, an additional request for the resource available from the content delivery network;***

Newton-471 explains the edge server may “cache popular content to serve the content in response to a future request” (i.e., additional request for the resource). EX-1002, ¶[0095]; *id.*, ¶[0059] (explaining “once content has been cached at the edge server, the edge server” will notify “the popularity service every time it gets another request for that content” and will determine “it has the requested content”), claim 10; EX-1006, ¶¶126-128.

- b. ***10(b): determining that the resource is cached at the first server of the first tier of servers; and***

As explained for limitation 10(a), Newton-471’s edge server will determine, in response to another request for content it has already cached, that it has the requested content in its cache. EX-1002, ¶¶[0048], [0059], [0095]-[0096], claim 10; EX-1006, ¶¶129-131.

- c. ***10(c): providing the cached resource from the first server of the first tier of servers in response to the additional request for the resource.***

As explained for limitation 10(a), Newton-471’s edge server will determine, in response to another request for content it has already cached, that it has the

requested content and will serve that content. EX-1002, ¶¶[0059], [0095]-[0096], claim 10; EX-1006, ¶¶132-134.

12. Independent Claim 11

a. *11(preamble): A content delivery network comprising:*

Newton-471 discloses this limitation for the same reasons provided above for limitation 1(p). *Supra* §VI.A.2.a; *see also* EX-1002, claim 11; EX-1006, ¶135.

b. *11(a): a first tier of servers comprising a first plurality of servers, an edge server of the first tier of servers receiving a request from a requesting device in communication with the first server for a resource available from the content delivery network;*

Newton-471 discloses this limitation for at least the same reasons as above for limitation 1(a). *Supra* §VI.A.2.b; *see also* EX-1002, claim 11; EX-1006, ¶136.

c. *11(b(i)): a second tier of servers comprising a second plurality of servers,*

Newton-471 discloses this limitation for at least the same reasons as above for limitation 1(c) and claim 9. *Supra* §§VI.A.2.d, VI.A.10; *see also* EX-1002, claim 11; EX-1006, ¶¶137-138.

- d. ***11(b(ii)): a first server of the second plurality of servers receiving a request from the edge server for the content and, in response, transmitting a redirect instruction intended for the requesting device to the edge server to obtain the resource from a content server and transmitting an instruction to not cache the portion of the resource at the edge server when the portion of the resource is obtained from the content server; and***

Newton-471 discloses this limitation for at least the same reasons as above for limitations 1(c)-1(e). *Supra* §§VI.A.2.d-f; *see also* EX-1002, claims 11, 12; EX-1006, ¶¶139-140.

- e. ***11(c): a popularity service tracking a popularity designation associated with the requested resource;***

Newton-471 discloses this limitation for at least the same reasons as above for limitation 1(b). *Supra* §VI.A.2.c; *see also* EX-1002, claim 11; EX-1006, ¶¶141-142.

- f. ***11(d): wherein the edge server processes the redirect instruction to obtain the content from the content server and provides the obtained content to the requesting device.***

Newton-471 discloses this limitation for at least the same reasons as above for limitations 1(d)-1(f). *Supra* §§VI.A.2.e-g; *see also* EX-1002, claim 11; EX-1006, ¶¶143-144.

13. Claims 12-18

Newton-471 discloses these claims for at least the same reasons as above for claims 2-8, respectively. *Supra* §§VI.A.3-9; *see also* EX-1002, claims 12-18; EX-1006, ¶¶145-161.

14. Independent Claim 19

- a. ***19(preamble)-19(c): A method of content delivery in a content delivery network comprising: receiving, at a first server of a first tier of servers of the content delivery network, a request from a requesting device for a resource available from the content delivery network; accessing a popularity service associated with the first server of the first tier of the content delivery network to determine a popularity designation associated with the requested resource; receiving an instruction to not cache the resource at the first server of the first tier of servers when the portion of the resource is obtained from the content server of the content delivery network;***

Newton-471 discloses these limitations for at least the same reasons as above for 1(preamble), and limitations 1(a), 1(b), and 1(e), respectively, and, Newton-471 also discloses limitation 19(b) for the additional reasons above for claim 8. *Supra* §§VI.A.2.a-c, VI.A.2.f, VI.A.9; *see also* EX-1002, claim 19; EX-1006, ¶¶162-169.

- b. ***19(d): requesting the resource from a second server of the content delivery network based at least on the popularity designation associated with the requested resource; and***

Newton-471 discloses this limitation for at least the same reasons as above for limitations 1(b)-1(c), claim 2, and limitation 3(a). *Supra* §§VI.A.2.c-d, VI.A.3-

4.a; EX-1002, claim 19. For instance, Newton-471 provides that the request to a second server of the CDN is based at least on whether the popularity designation is unpopular, popular, or somewhere in-between. EX-1002, ¶¶[0054]-[0057], [0063], Fig. 5; EX-1006, ¶¶170-171.

c. *19(e): providing the obtained resource to the requesting device.*

Newton-471 discloses this limitation for at least the same reasons as above for limitation 1(f). *Supra* §VI.A.2.g; EX-1002, claim 19; EX-1006, ¶¶172-173.

15. Claim 20: “*The method as recited in claim 19 wherein the second server of the content delivery network is a content server storing the requested content.*”

Newton-471 explains the content server may be an edge server (i.e., a peer server of the first server). EX-1002, ¶¶[0090] (“content server (e.g., edge server)”), ¶[0091] (“content server may be an edge server”), claim 20. Thus, Newton-471 discloses this limitation for the same reasons as above for claims 5 and 15. *Supra* §§VI.A.6, VI.A.13. Alternatively, Newton-471 explains “the content server may be a device in one of the other tiers discussed above, such as a parent server or origin server.” EX-1002, ¶[0091]. Thus, where the content server is a parent server, Newton-471 discloses this limitation for the same reasons as above for limitation 1(c), claim 4, limitation 11(b)(ii), claim 14, and limitation 19(d). *Supra* §§VI.A.2.d, VI.A.5, VI.A.12.d, VI.A.13, VI.A.14.c. Newton-471 also discloses this limitation where the content server is an origin server because, in Newton-471’s exemplary

single tier CDN, the content server would be the origin server. EX-1002, ¶¶0034]; EX-1006, ¶¶174-176.

B. Ground 2: Dilley, Pai, and Wang Render Obvious Claims 1-20

1. Overview of Dilley

Dilley discloses a CDN using a tiered cache hierarchy to efficiently distribute content. EX-1003, Abstract; EX-1006, ¶¶177-183. Dilley's tiered hierarchy of servers includes edge, parent, and origin servers. EX-1003, 2:32-36, 3:67-4:2, 4:63-5:15. The system receives a request and routes it to the nearest edge server. EX-1003, 7:2-6, 8:3-10. If the edge server has the requested content, it will directly serve that content to the user. EX-1003, 8:3-10. Otherwise, the edge server will request the content from another server, e.g., a peer, parent, or origin server. EX-1003, 8:11-48.

Content is cached and fetched from the different server levels based on criteria defined by, e.g., content metadata. EX-1003, 6:10-40, 7:6-40. Metadata tags may be used to control how content is distributed and from which server/tier it should be served. *Id.*; EX-1003, 4:23-26, 6:4-15, 8:23-48, 9:57-62. The origin server is contacted only when necessary to ensure it is not overwhelmed. EX-1003, 1:8-10, 2:32-36, 2:67-3:2, 5:65-6:15, 9:57-65.

Dilley teaches that its tag-based approach provides flexibility depending on the needs of the content provider and allows for content to be organized based on

metadata attributes specified by the content provider. EX-1003, 6:10-15, 8:23-48.

It also teaches its system insulates the origin server from requests for popular content, thereby improving CDN performance. EX-1003, 1:8-10, 2:12-31.

2. Overview of Pai

Pai discloses a system and method for popularity-based on-demand media distribution. EX-1004, Abstract; EX-1006, ¶¶184-188. Pai's design facilitates media storage and delivery to users by adjusting content storage locations in a network environment based on content popularity. EX-1004, ¶¶[0013], [0016].

Pai assigns popularity values to each content item and regularly updates those values. EX-1004, ¶[0026]. Pai dynamically redistributes content between different storage levels (i.e., level 1, 2, and 3) on the network based on popularity thresholds. EX-1004, ¶¶[0018], [0038]-[0045].

Pai explains that its approach ensures that more frequently accessed content is readily available and able to be delivered quickly, while less popular content can be stored further from users, occupying less valuable storage capacity (i.e., slower, long-term storage options). EX-1004, ¶¶[0017], [0020].

3. Overview of Wang

Wang discloses a tiered service network having service managers responsible for each different tier of the network. EX-1005, Abstract, ¶[0022]. Wang explains when a request is received, the system will check the first level cache to determine

if the content is available. EX-1005, ¶[0123]. If the content is not found, the request escalates up the service hierarchy from level 1 to level 2 to level 3 (if the content is not available at level 2). *Id.* Wang identifies items that should not be cached by the lower-level service manager based on cache control attributes. EX-1005, ¶[0123].

A POSITA would have understood that identifying items to not cache would improve system performance by avoiding the caching of stale, private, or unpopular content that would otherwise take up valuable storage space. EX-1006, ¶¶189-191.

4. Motivation to Combine

The '573 patent “relates to content delivery, to content delivery networks (CDNs), and to frameworks and systems using CDNs.” EX-1001, 1:46-48. Dilley, Pai, and Wang are each similarly directed to networks for storing and delivering content. *See* EX-1003, Abstract; EX-1004, Abstract; EX-1005, Abstract. Dilley, Pai, and Wang are also directed to the same problem as the '573 patent—efficient and reliable management of network resources. *See* EX-1001, 10:5-41 (describing efficiency aims); EX-1003, 2:26-31 (CDN to provide “desired content quickly and reliably”); EX-1004, ¶[0013] (popularity-based distribution to improve network); EX-1005, ¶[0003] (multi-level network to improve “server congestion and Internet traffic jams”). A POSITA concerned with network management would have been motivated to combine Dilley, Pai, and Wang. EX-1006, ¶¶192-194.

a. *Combining Dilley and Pai*

First, a POSITA would have found it obvious to combine Dilley (EX-1003) with Pai (EX-1004) to provide an improved and dynamic tiered CDN with a popularity service. EX-1006, ¶¶195-198. Dilley teaches that its tiered cache hierarchy, which stores more frequently accessed content closer to the requesting clients (e.g., edge servers) and less frequently requested content in further caches (e.g., parent servers or origin servers), improves content retrieval time for popular content while providing cost effective storage options for the content provider. EX-1003, 2:12-17, 2:23-25, 2:63-3:2, 5:65-6:10. Dilley determines where to cache content “as [a] function of metadata associated with the given object request.” EX-1003, 2:42-56.

A POSITA would have understood that using “object specific metadata” to facilitate control over a tiered distribution of content in a CDN could be done using content popularity. EX-1003, 2:23-25. Indeed, Dilley at least suggests using the popularity of objects to distribute content within a CDN and “reduce traffic to the origin servers for very popular” objects. EX-1003, 1:8-10, 2:26-31; *see also* EX-1003, 4:47-53, Fig. 2 (describing prior art configuration with “hot” object cache); EX-1006, ¶¶199-200.

Pai further describes such a popularity-based system. EX-1004, ¶¶[0003], [0013], [0029]. It teaches how to distribute more and less popular content between

different cache tiers and how to determine popularity and dynamically redistribute content. *Id.*; EX-1006, ¶¶202-207.

In view of Dilley's suggestion, a POSITA would have been motivated to implement Pai's more specific teachings by providing dynamic popularity-based media distribution in Dilley. *Optivus Tech., Inc. v. Ion Beam Applications S.A.*, 469 F.3d 978, 991 (Fed. Cir. 2006) (finding motivation to combine exists as a matter of law where one reference specifically addresses subject matter of another reference). Combining Dilley's tiered cache architecture with Pai's dynamic popularity distribution would have led to a more efficient, scalable, and cost effective CDN. EX-1003, 1:30-32 (explaining an effective CDN serves frequently accessed content from an optimal location), 2:26-31 (describing quick and reliable content distribution without overwhelming the origin server with requests for popular content); EX-1006, ¶¶196, 201, 206-209. A POSITA would have understood that this combination would allow content to be placed dynamically in the appropriate cache, based on its popularity, and would improve cache management, delivery speed, and resource optimization. EX-1003, 5:16-21, 5:65-6:10 (noting congestion benefits of tiered structure); EX-1004, ¶[0002] (explaining cost benefits of optimized distribution); EX-1006, ¶¶196, 209-211. A POSITA would have also understood that the combined system would have allowed for improved scalability of the CDN because it would combine Dilley's existing tiered hierarchy with Pai's dynamic

popularity-based distribution system, which would account for evolving media consumption patterns. EX-1006, ¶¶210-211.

The combination would have been nothing more than combining known prior art elements according to understood principles to yield predictable results. EX-1006, ¶¶208-210. Both Dilley and Pai describe networks having many common features. For example, each uses a hierarchical server structure to deliver content to clients such that when a requested object is not available from a first content server, the object may be retrieved from another server or tier. EX-1003, 5:4-21; EX-1004, ¶¶[0013], [0017], [0039]. Given the similarities and overlap in Dilley's and Pai's teachings, a POSITA would have had a reasonable expectation of success in making the combination. EX-1006, ¶¶211-212.

b. *Further Combining Dilley and Pai with Wang*

Second, a POSITA would have found it obvious to combine Dilley and Pai with Wang to provide a system with, in certain situations, instructions to not cache a resource at the edge when it is obtained from the content server. EX-1006, ¶¶213-216.

Like Dilley and Pai, Wang provides for a tiered network. EX-1005, Abstract, ¶[0005] (describing a hierarchical service and content distribution system). And like Dilley and Pai, Wang also discloses a three-tier network where each level may store content of different popularity thresholds. EX-1005, ¶¶[0047]-[0049]. Wang

teaches that such a CDN can be improved by including a mechanism to identify certain resources that should not be cached by the network service engines. EX-1005, ¶[0123].

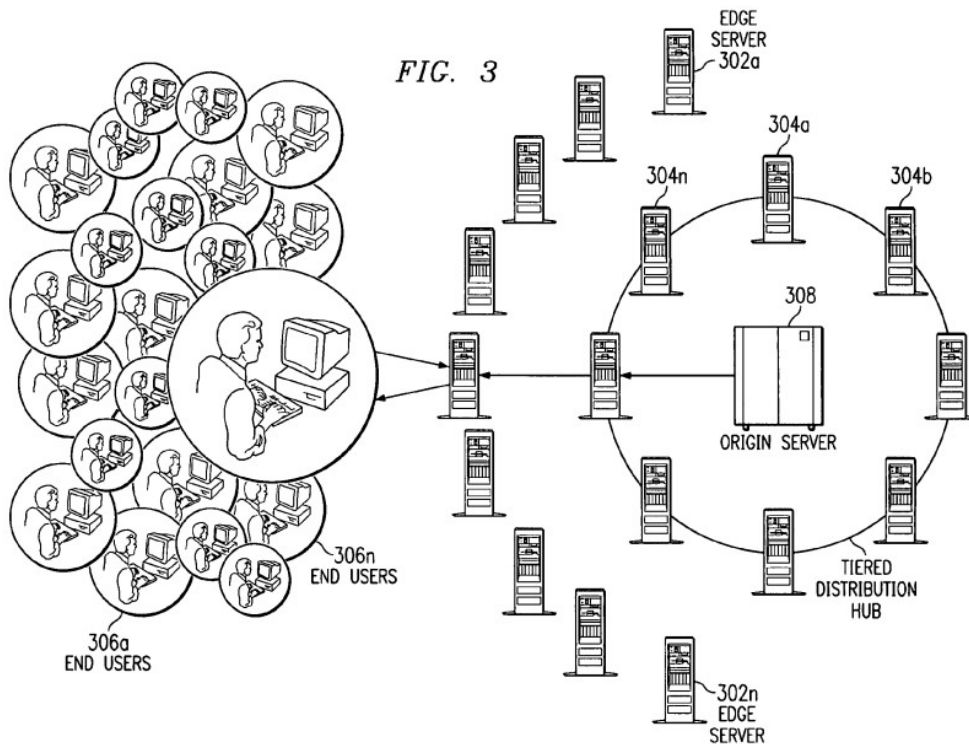
A POSITA would have been motivated to combine this feature with Dilley and Pai to avoid the cost and resource expenditure of caching content that, e.g., becomes stale. EX-1006, ¶¶217-218 (citing EX-1013, ¶[0011] (explaining the infeasibility of storing all content on edge servers). A POSITA would have understood that implementing Wang’s teachings would have improved system operation. *Id.*

Dilley similarly teaches that certain “un-cacheable” content should be served from the origin server, EX-1003, 2:12-17, and both Dilley and Pai describe a similar tiered cache hierarchy. EX-1003, Fig. 3; EX-1004, Fig. 2. Thus, combining Wang’s instruction to not cache a resource at the edge would have been nothing more than combining known features in the art according to understood principles to yield a predictable result—a tiered CDN with a hierarchical cache that escalates a request up the chain (edge to parent to origin) to find content and that instructs lower-level servers to not cache that content if that content fails to meet certain criteria. EX-1004, Fig. 6. A POSITA would have reasonably expected success in making the combination, for identifying cacheable or uncacheable content was well known and understood in the art. EX-1006, ¶219.

5. Independent Claim 1

- a. *1(preamble): A method of content delivery in a content delivery network comprising:*

Dilley is entitled “[m]ethod and system for tiered distribution in a content delivery network,” and further discloses “[a] tiered distribution service” provided “in a content delivery network.” EX-1003, cover, 1:12-40.

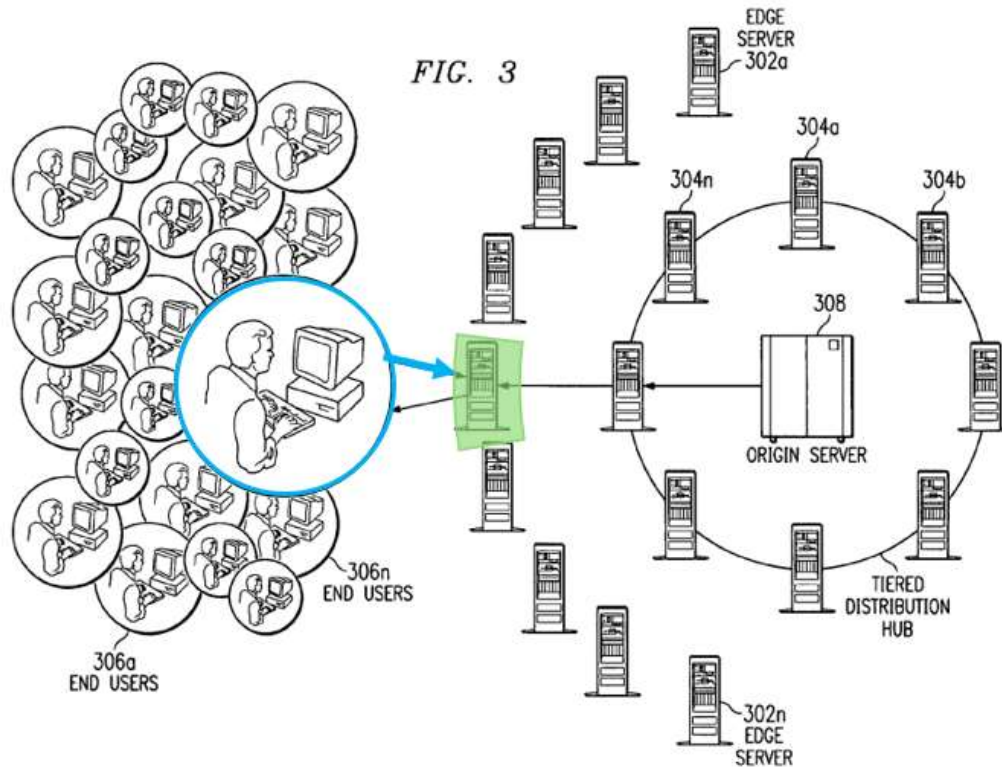


EX-1003, Fig. 3 (depicting tiered distribution service in CDN); EX-1006, ¶¶220-223.

- b. *1(a): receiving, at a first server of a first tier of servers of the content delivery network, a request from a requesting device for a resource available from the content delivery network;***

Dilley's CDN employs a tiered distribution structure, where content is cached and served to requesting users from edge servers (a first tier of servers), as well as parent regions, and, when necessary, directly from the origin server. EX-1003, 4:63-5:21.

As an initial step, an end user (requesting device) makes a content request that is received by an edge server (a first server) of the CDN. EX-1003, 4:63-5:21. Dilley discloses the edge server "receives end-user requests for origin server content and serves this content" from its cache "if it can do so." EX-1003, 7:50-53. This is shown in Dilley Figure 3:



EX-1003, Fig. 3 (annotated, showing edge server receiving request from end user), Fig. 6, 8:3-7 (explaining “when a given edge server receives a client request for an object[, t]he routine begins”).

Dilley also discloses that when a request is made by a user, and received by an edge server, the edge server of the CDN will serve or otherwise attempt to obtain the resource from the CDN, confirming the requested resources are available from the CDN. EX-1003, 8:3-21; EX-1006, ¶¶224-230.

- c. ***1(b): accessing a popularity service associated with the content delivery network to determine a popularity designation associated with the requested resource;***

As explained, *supra* §VI.B.4, while Dilley teaches the advantages of using a hierarchical infrastructure to distribute popular content closer to the edge, it leaves to a POSITA how the popularity service would work in application, and a POSITA would have been motivated to implement Pai's teachings regarding its popularity service in Dilley's system with a reasonable expectation of success. EX-1006, ¶¶231-236.

Pai's popularity service monitors and calculates a content's popularity and, based on this calculated popularity value, dynamically adjusts the content's storage location to improve content delivery to users while efficiently managing the system's limited storage capacity. EX-1004, ¶¶[0013], [0020]. For instance, Pai's system tracks the requests made by users, and accordingly updates popularity values for requested content, potentially triggering a redistribution to another level. EX-1004, ¶¶[0039], [0047].

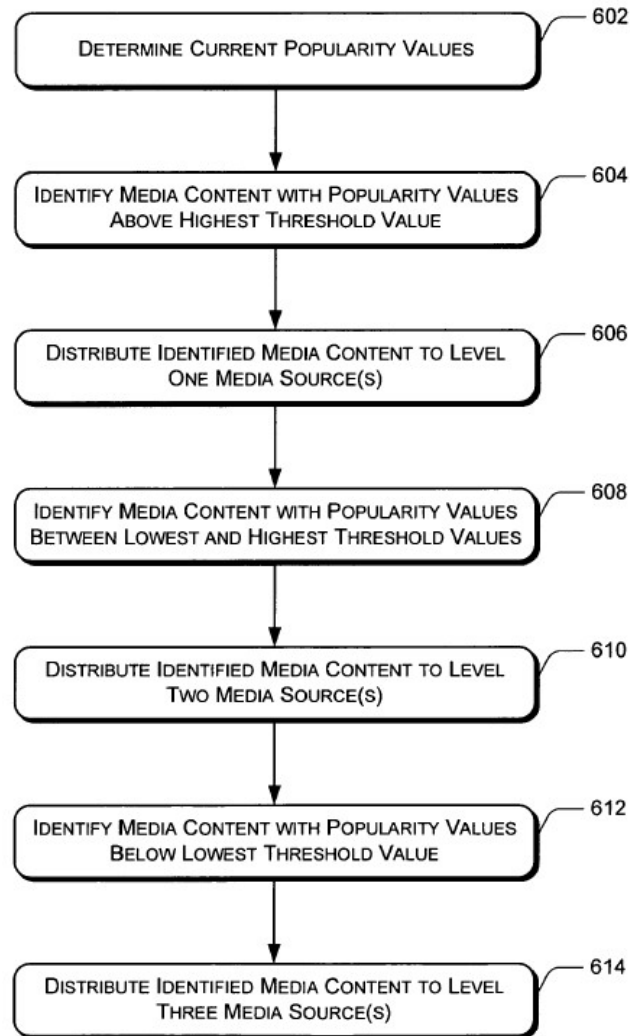


Figure 6

EX-1004, Fig. 6.

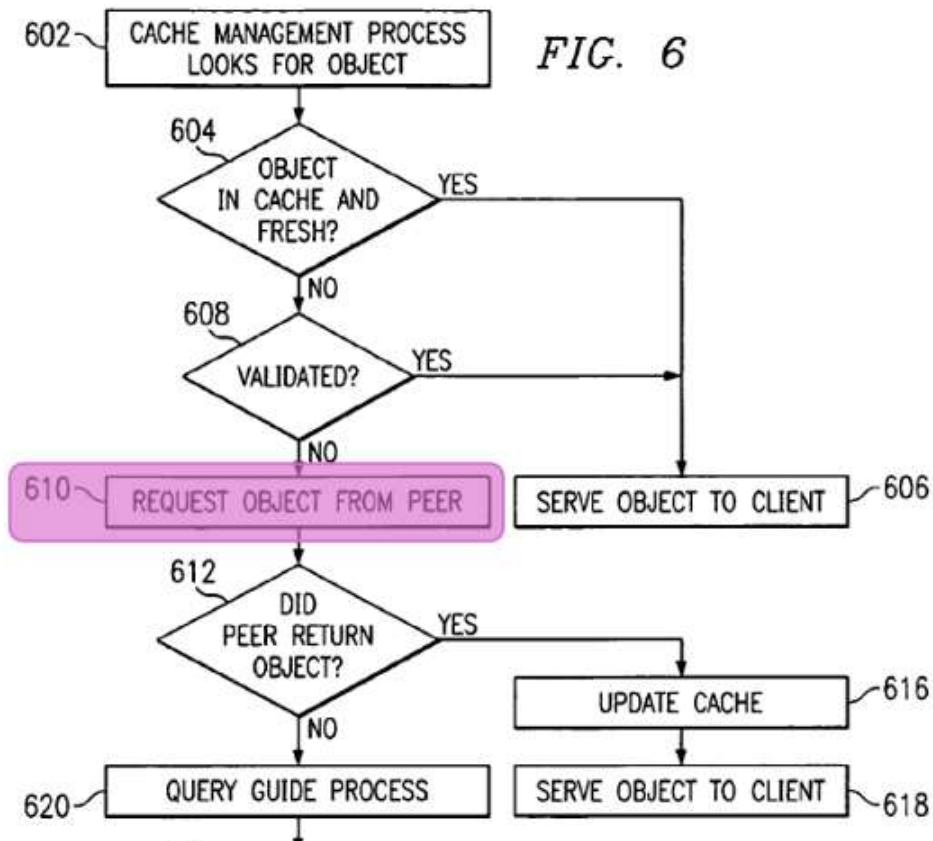
Through monitoring and dynamically adjusting a content's popularity value, Pai discloses accessing a popularity service to determine the popularity associated with the requested resource. EX-1004, ¶¶[0039], [0047]. Indeed, content in Pai is appropriately distributed throughout the system based on popularity such that when a user requests a resource, the system retrieves that resource from the relevant media

source based on the content's current storage location (which is selected based on current popularity). *See e.g.*, EX-1004, ¶¶[0039]-[0045]; EX-1006, ¶¶237-241.

d. *1(c): requesting the resource from a second server of the content delivery network;*

The combination discloses this limitation in at least two ways. EX-1006, ¶242; *see also* §VI.B.4 (explaining motivation to combine with expectation success).

First, Dilley explains when a request is received at an edge server, the server first looks for the object in the edge server cache (step 604). EX-1003, 8:5-11. If present, the edge server will serve the object to the client. *Id.* Otherwise, the edge server will request the object from a different server (step 610). EX-1003, 8:11-18. For instance, “[i]f the object is not in cache or needs to be validated, the edge server makes a request (e.g., via a cache sharing protocol) to other edge servers in the region (each a ‘peer’) to see if one of them has a fresh copy of the object.” EX-1003, 8:3-15.

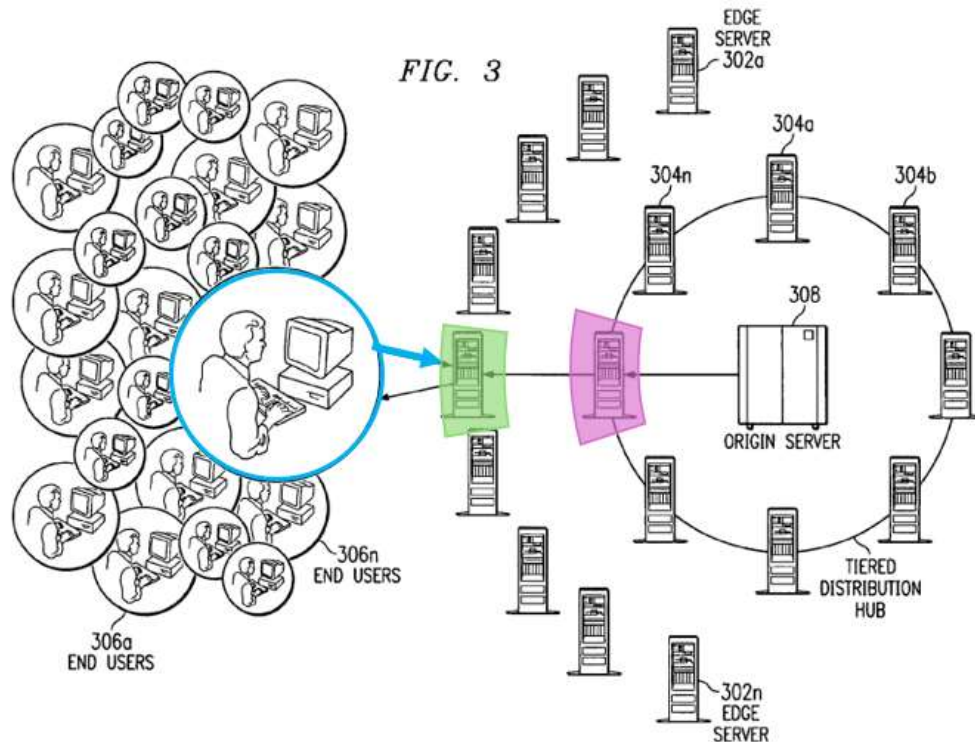


EX-1003, Fig. 6 (abbreviated, annotated).

Thus, Dilley discloses that the first server (i.e., edge server receiving request) will request that resource from a second server (i.e., peer server in the same tier) when it does not have the requested content. EX-1003, 5:22-30, 8:11-18; EX-1006, ¶¶243-245.

Second, Dilly also discloses using “a cache hierarchy[] in which requests for content that cannot be serviced in edge servers are funneled through intermediate nodes,” i.e., requesting a resource from a second server of the CDN. EX-1003, 2:5-17. As Dilley explains, its CDN comprises an “edge server region” of edge servers

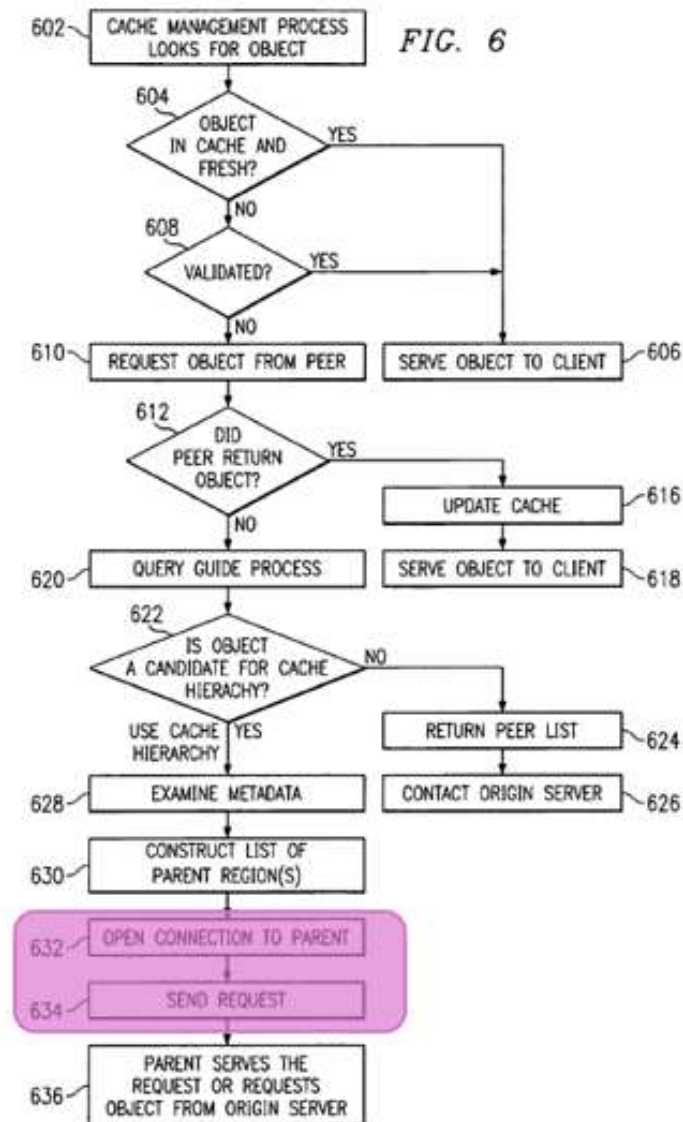
302a-n, and a single parent region or a subset of parent server regions containing parent servers 304a-n. EX-1003, 2:37-46; EX-1006, ¶246.



EX-1003, Fig. 3 (annotated).

When the edge server cannot service the request, “instead of contacting the origin server, the request is provided to either the single parent region” or to one of the subsets of parent server regions “for handling,” (i.e., a second server of the CDN). EX-1003, 2:46-56. In this instance, the request is serviced, if possible, by the second server (the parent server), and “is only forwarded on to the origin server if it cannot be serviced” by the parent server. EX-1003, 2:52-56; *see also id.*, 5:27-41.

This is illustrated in Figure 6, where an edge server, without the requested content, makes a request to another edge server in the region (i.e., a peer server) to see if a copy of the content is available (step 610). EX-1003, 8:7-15. If the peer server is also unable to return the object, Dilley will communicate the request to a parent server of the CDN (steps 632 and 634). EX-1003, 8:38-48 (explaining edge server “sends the request to a server in the parent region).



EX-1003, Fig. 6 (annotated); EX-1006, ¶¶247-249.

- e. ***1(d): processing, at the first server of the first tier of servers, a redirect instruction from the second server of the content delivery network to obtain the resource from a content server of the content delivery network;***

If Dilley's edge server does not have the requested content available, it will try to request that content from another server, e.g., a peer server or a parent server. EX-1003, 5:4-21, 7:31-40. This second server will either serve the content from its cache or instruct the first server to request the content from a server higher on the hierarchy, such as an origin server. EX-1003, 7:31-40, 8:53-58.

A POSITA would have understood that to implement the described process, when the second server is a parent server, Dilley would send a redirect instruction to the edge server, directing the edge server to obtain the resource from the origin server instead. EX-1006; ¶¶250-253; EX-1003, 7:15-30. Indeed, a POSITA would have been motivated to implement this redirect instruction from Wang. *See e.g.*, EX-1005, ¶¶[0123]-[0130]; EX-1006, ¶¶253-254.

Wang discloses a CDN managed hierarchically into three tiers (levels 1, 2, and 3), where each level supports content caching. EX-1005, ¶¶[0047]-[0049]. Wang explains that information is distributed from lower levels (level 1) to higher levels (level 3) to promote efficiency and avoid conflicts. EX-1005, ¶[0046]. For instance, when a user requests content, the system will select a level 1 server to

retrieve the content. EX-1005, ¶[0123]. If the content is not found at level 1, the request escalates from level 1 to level 2 and eventually to level 3 to find the content. *Id.* In this instance, Wang explains that the level 1 server will process a redirect instruction to obtain the content from the origin server. EX-1005, ¶[0123] (explaining, for not-cacheable content, the level 1 server will redirect the request to the origin server), ¶[0124] (explaining the same redirect process to the origin server is supported for cacheable content).

A POSITA would have understood that these redirects were instructions to obtain the content from the origin content server. EX-1006, ¶255. And a POSITA would have understood that Wang's redirect procedure would have added benefits because the edge server would then process the redirect instruction. It would thus be expecting the response directly from the origin server. Such a design would decrease latency as it would remove the parent server as a "middleman," and would allow the edge server to communicate with and receive content directly from the origin server. EX-1006, ¶¶255-256. In addition, this process would allow the parent server to remain available to service other requests received by the CDN. EX-1006, ¶257.

Each of the above benefits would apply equally to the example where the second server is a peer server such that a POSITA would recognize the benefits of having the peer server sending a redirect instruction to the edge server, directing the

edge server to obtain the resource from a parent or origin content server instead. EX-1006, ¶258.

A POSITA would have understood having the second server (either a peer or parent server) send a redirect instruction to the edge server to receive the content from a content server would have been one of very finite design choices and would have been obvious in the context of at least Dilley and Wang. EX-1006, ¶¶259-260.

- f. ***1(e): receiving an instruction to not cache the portion of the resource at the first server of the first tier of servers when the portion of the resource is obtained from the content server of the content delivery network; and***

As explained, *supra* §VI.B.4.b, Dilley and Pai teach the advantages of using a hierarchical infrastructure to distribute popular content closer to the edge, and a POSITA would have been motivated to implement Wang's instructions to not cache a resource at the edge when it is obtained from the content server in Dilley-Pai with a reasonable expectation of success. EX-1006, ¶¶261-266.

As explained above, in Wang's system, when a user requests content, the system will select a level 1 server to retrieve the content. *Supra* §VI.B.5.e; EX-1005, ¶[0123]. If the content is not found at the level 1 server, the request escalates from level 1 to level 2 and eventually to level 3. *Id.*

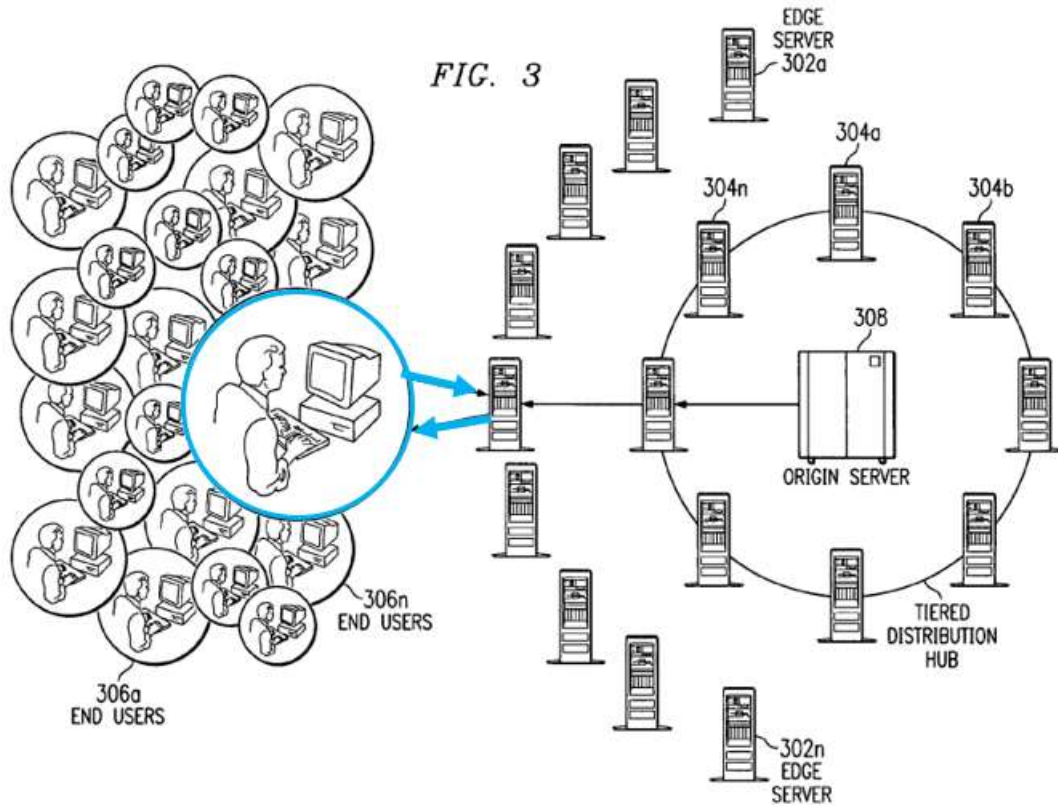
Wang discloses that content items may include cache attributes identifying whether they should be cached or not. EX-1005, ¶[0123]. The system can include

content attributes that identify the content as not cacheable. *Id.* Specifically, Wang discloses that when the content is unavailable at levels 1 and 2, and is obtained from level 3, it may include an “indication” that it is not to be cached (i.e., a no-cache instruction). *Id.* A POSITA would have understood that this no-cache attribute attached to the content’s metadata would be an instruction for the level 1 server to not cache the resource at level 1 when it is obtained from the level 3 server (i.e., origin or content server) that is received at the level 1 (edge) server with the resource. EX-1006, ¶¶267-268.

As above, a POSITA would have understood the no-cache instruction ensures not every item is cached at the lower levels. EX-1006, ¶269. A POSITA would have also understood certain items should not be cached at edge servers, including not only dynamic or private content but also unpopular content in the context of Dilley, Pai, and Wang, where storage capacity at the edge is more limited. EX-1006, ¶270.

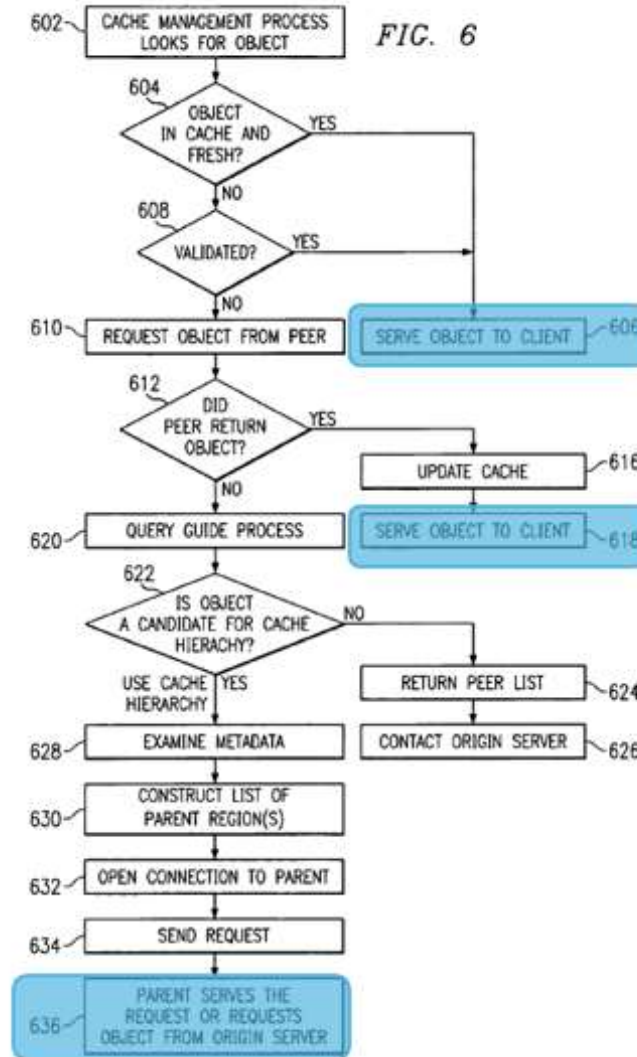
g. *1(f): providing the obtained resource to the requesting device.*

Dilley’s overall purpose is to distribute content to end-users. It uses a tiered cache hierarchy in a CDN to do so. EX-1006, ¶¶271-274.



EX-1003, Fig. 3 (annotated).

While Dilley's process begins with an end-user making a request for content, it ends with that requested content being served to the client.



EX-1003, Fig. 6 (annotated) (steps to “serve object to client”), 8:3-18, 8:53-58.

6. **Claim 2:** “*The method as recited in claim 1 wherein the popularity designation associated with the requested resource indicates that the requested resource is not popular.*”

Dilley discloses a system using a cache hierarchy to store and serve popular objects from the edge, such “that the [origin] site is not overwhelmed with requests for popular content.” EX-1003, 5:16-21. For instance, Dilley’s “[c]ache hierarchy

uses a hierarchy of servers to distribute content closer to the requesting end users,” resulting in a “particularly advantageous [functionality] for flash-popular content.” EX-1003, 9:57-63. Thus, a POSITA would have understood that Dilley determines and indicates if a requested resource is not popular, and thus can be served from a higher tier server, such as the origin server. EX-1006, ¶¶275-276.

Moreover, Pai’s popularity service specifically identifies whether a resource is not popular. Pai’s popularity-based media distribution system distributes popular content from one location and less popular content from another. *See* EX-1004, Summary. To do this, Pai determines popularity values of content and identifies: (1) content with popularity values above a highest threshold (step 604), (2) content with popularity values between lowest and highest threshold values (step 608), and (3) content with popularity values below a lowest threshold (step 612). EX-1004, Fig. 6:

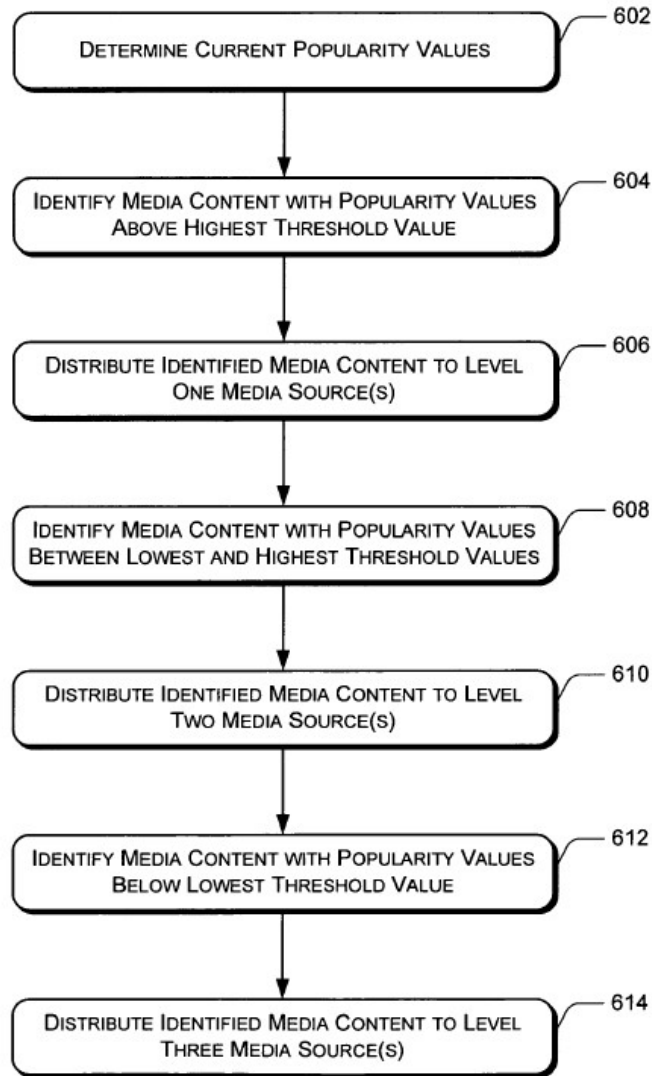


Figure 6

The media content identified as less popular will be maintained by a media source located physically further from the requesting devices. EX-1004, ¶¶[0013], [0016] (“less popular media content may be served from media sources associated with another level” further from requesting devices), [0017], [0020] (“level three media source 216 is configured to maintain and serve the least popular media content”). Pai explains, as an example, that these unpopular items could be those

with popularity values below the lowest popularity threshold, such as those assigned a value between one and two (on a scale of one to ten). EX-1004, ¶[0044] (“media content with popularity values below the lowest popularity threshold is identified”); EX-1006, ¶¶277-278.

Pai also illustrates distributing media content based on popularity in Figure 3 below:

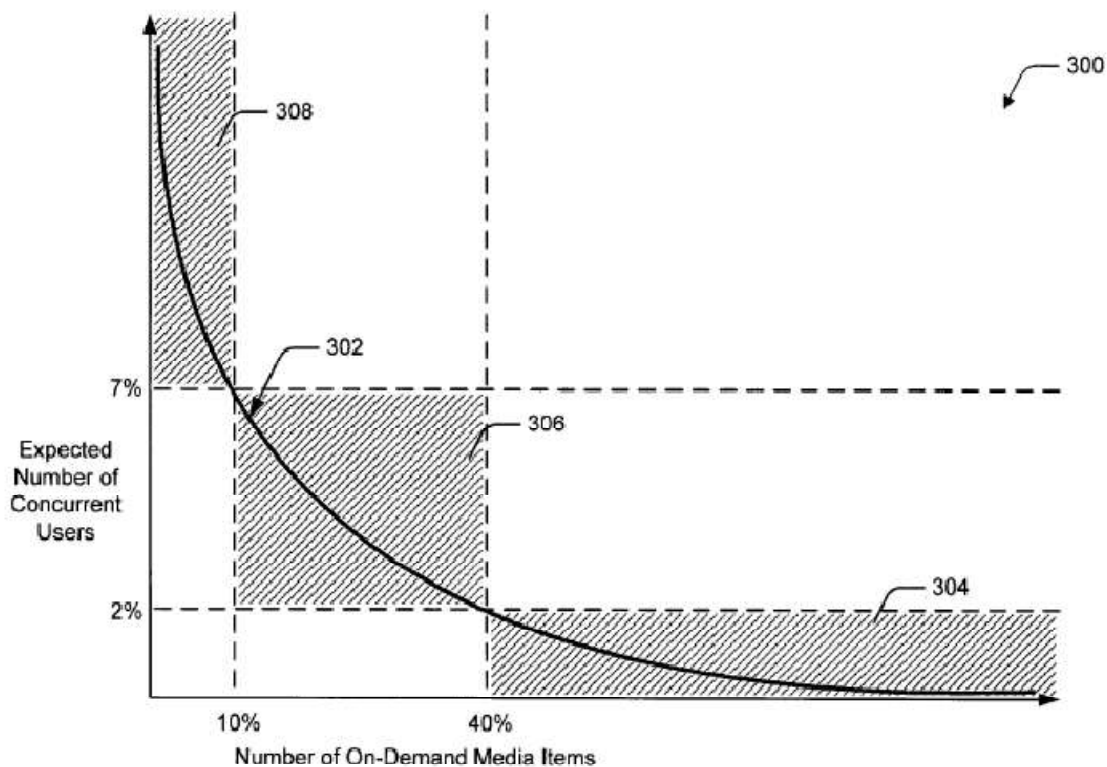


Figure 3

EX-1004, Fig. 3. Pai explains it can use the determined number of concurrent users of a resource to determine resource popularity. EX-1004, ¶¶[0021]-[0023], [0039]. As one example, Pai explains that the shaded area 304 represents the largest number

of resources that have the fewest number of concurrent users (i.e., least requested content), and thus “represents the least popular media content.” EX-1004, ¶[0021].

As discussed above, *supra* §VI.B.4.a, a POSITA would have been motivated to implement Pai’s popularity service in Dilley with a reasonable expectation of success. EX-1006, ¶¶279-280.

7. Claim 3

- a. ***3(a): The method as recited in claim 1 wherein the popularity designation associated with the requested resource indicates that the requested resource is popular,***

Dilley’s system uses a cache hierarchy to store and serve popular objects from the edge, such “that the [origin] site is not overwhelmed with requests for popular content.” EX-1003, 5:16-21. For instance, Dilley explains that the “[c]ache hierarchy uses a hierarchy of servers to distribute content closer to the requesting end users,” resulting in a “particularly advantageous [functionality] for flash-popular content.” EX-1003, 9:57-63. Thus, a POSITA would have understood that Dilley determines and indicates whether a requested resource is popular, and thus can be served from the edge servers, protecting the origin server from these requests for popular content. EX-1006, ¶281-282; EX-1003, 5:65-6:7.

Pai further discloses the popularity designation associated with the requested resource indicates it is popular. *See, e.g.*, EX-1004, ¶[0013]. Pai not only identifies popularity values of content below a lowest threshold value (least popular content),

but also identifies content with popularity values above a highest threshold value (indicating popular resource). EX-1004, ¶¶[0040]-[0041] (“media with popularity values above a highest popularity threshold are identified” and distributed to level one media sources); *see also id.*, ¶¶[0023], [0038]-[0047], Figs. 3 and 6 (identifying that a resource is popular at step 604 as shown above).

For instance, Pai discloses that “level two media sources 220 and 224 are configured to maintain and serve more popular media content,” while “level one media sources 218 and 222 are configured to maintain and serve the most popular media content.” EX-1004, ¶[0020].

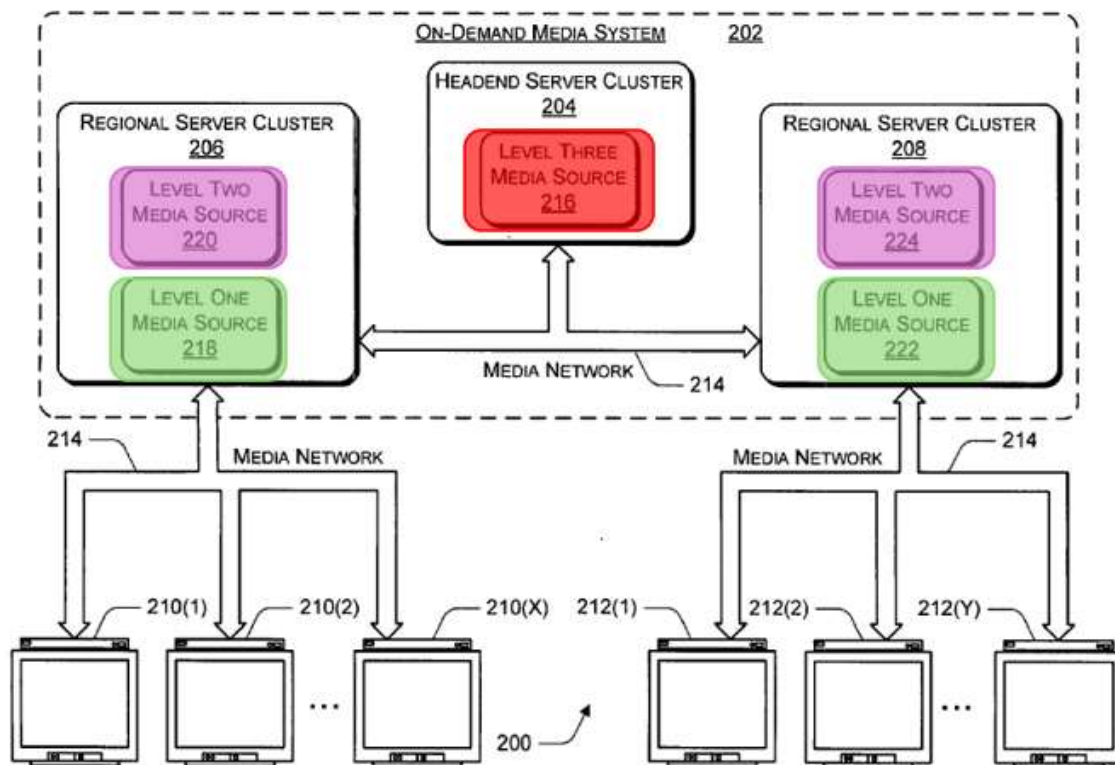


Figure 2

EX-1004, Fig. 2 (annotated).

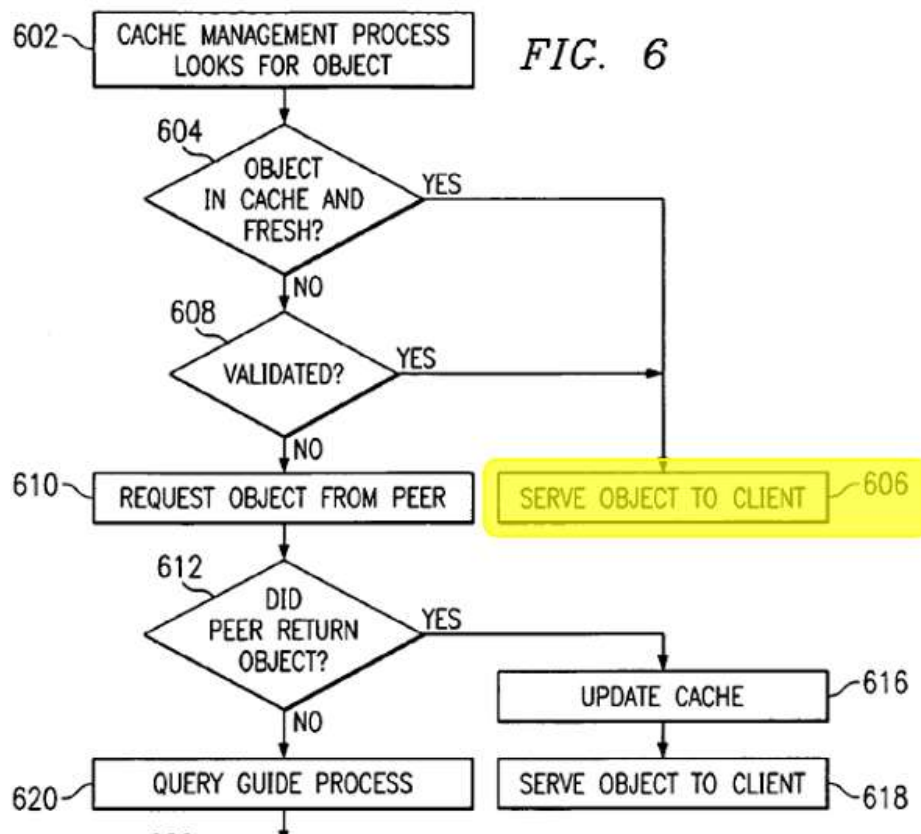
For the reasons discussed above, *supra* §VI.B.4.a, a POSITA would have been motivated to implement Pai's popularity service teachings with Dilley with a reasonable expectation of success. EX-1006, ¶¶283-287.

b. *3(b): the method further comprising: receiving an instruction to cache the resource at the first server of the first tier of servers when the resource is obtained from the content server of the content delivery network.*

The combination discloses receiving an instruction to cache the resource at the first server of a first tier of servers when a resource is obtained from the content server of the CDN.⁵ EX-1006, ¶¶288-298; *see also* §VI.B.4.a.

Because Dilley uses a cache hierarchy to store and serve popular objects from the edge, a POSITA would have understood that Dilley's servers send and receive instructions to cache popular content at the edge for future requests. EX-1003, 5:16-21, 9:57-63; EX-1006, ¶290. For instance, Dilley explains when a request is received at an edge server, it looks for the object in the edge server cache (step 604). EX-1003, 8:3-11. If it is there, it will serve the object to the client. *Id.* Otherwise, it will request the object from a different server (step 610), and update the cache to include the requested content. EX-1003, 8:11-21.

⁵ *Supra* §VI.A.4, n.4.



EX-1003, Fig. 6 (annotated and abbreviated).

A POSITA would have understood Dilley's system thus discloses receiving an instruction to cache items, since it will update the cache with the content it did not previously have. EX-1006, ¶291. A POSITA would have further understood that this would occur not only when content is received from peer servers, but also when popular content is retrieved from parent or origin servers, since that would in turn serve Dilley's stated goal of serving popular content from the edge and insulating the origin server from requests for such popular materials. EX-1003, 1:8-10, 2:26-31; EX-1006, ¶291.

Pai further confirms this understanding, and provides additional motivation for a POSITA to include an instruction to cache popular resources as claimed. EX-1006, ¶292. It discloses that its system will “dynamically redistribute media content” among different levels “based on popularity.” EX-1004, ¶[0053]. As Pai explains, a “[m]edia delivery engine 424 is configured to manage delivery of media content to client devices based on received requests for the media content,” (i.e., the number of requests for a given resource is used “to update popularity values associated with the media content, which may then be used to redistribute the media content across multiple levels of media sources”). EX-1004, ¶[0030]. Thus, Pai discloses “a server may, based on popularity, dynamically redistribute media content among multiple levels of media sources” so that the system will instruct the content to be cached at, e.g., the first server as it becomes more popular. EX-1004, ¶¶[0029], [0052]-[0054]; EX-1006, ¶¶292-294.

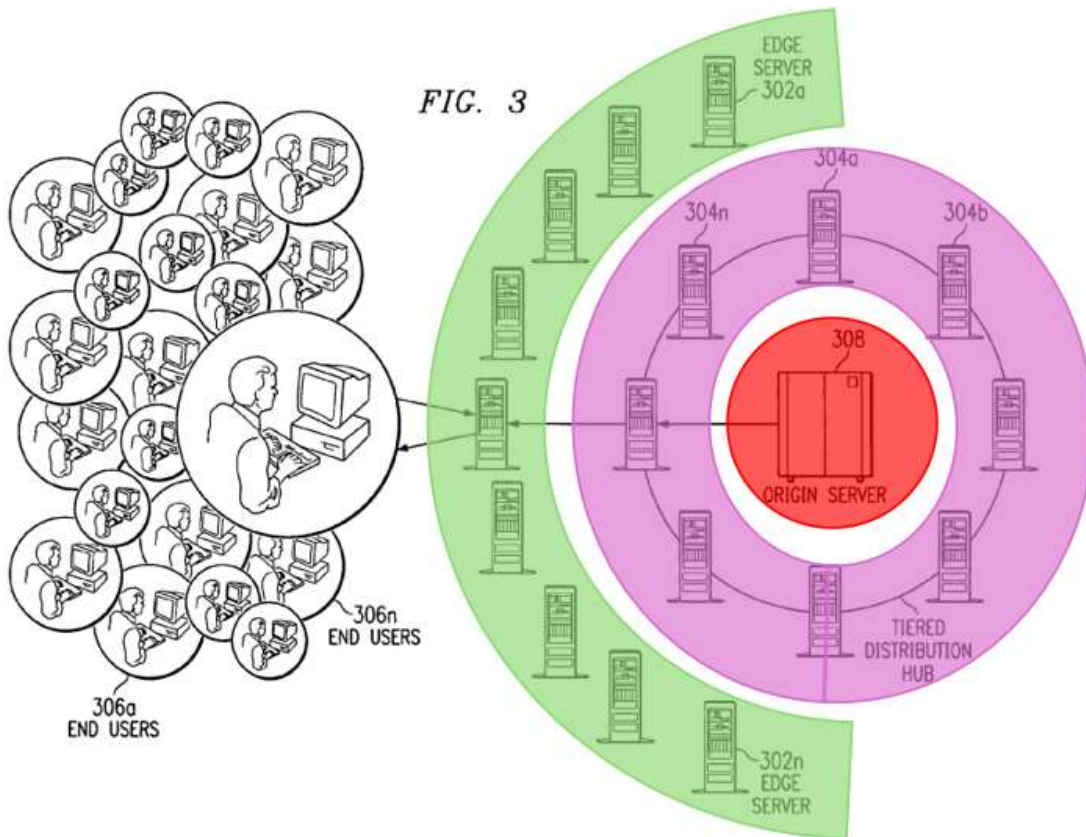
Wang similarly includes content-specific attributes, including an “indication of cacheable or not.” EX-1005, ¶[0123]. This includes not only an instruction not to cache content, but also an instruction to cache certain content. EX-1006, ¶295. For instance, “[i]f a cacheable content is indicated,” like a popular content, “cache the content if ‘cacheable’ is also specified in the http response from the origin server.” EX-1005, ¶[0124]. A POSITA would understand this cacheable attribute attached to the content’s metadata would be an instruction for the level 1 server to

cache the resource at level 1 when it is popular and obtained from the level 3 server (i.e., the origin server). EX-1006, ¶296. A POSITA would have understood that this cacheable instruction would help ensure that the popularly requested content is cached at the lower levels. EX-1006, ¶297.

The combined teachings of Dilley, Pai, and Wang, accordingly, disclose this limitation. EX-1006, ¶298. A POSITA would have been motivated to implement these teachings to serve Dilley's goals, EX-1003, 1:8-10, 2:26-31, and achieve the benefits described in Pai, EX-1004, ¶¶[0013], [0016]-[0017], and Wang, EX-1005, ¶[0123]. *See* §VI.B.4.a.

8. Claim 4: “*The method as recited in claim 1 wherein the second server is a metro mid-tier server of the content delivery network.*”

Dilley explains that the second server could be a parent server. *See* EX-1003, Fig. 4. A POSITA would have understood that a parent server, below in purple, is a mid-tier server as it sits in a middle tier between the edge and the origin. EX-1006, ¶¶299-300.



EX-1003, Fig. 3 (annotated).

The same is true for Pai's level two media source, which is effectively the parent server, sitting in a middle tier between the level one media source (e.g., edge server) and the level three media source (e.g., origin server). EX-1004, ¶[0020].

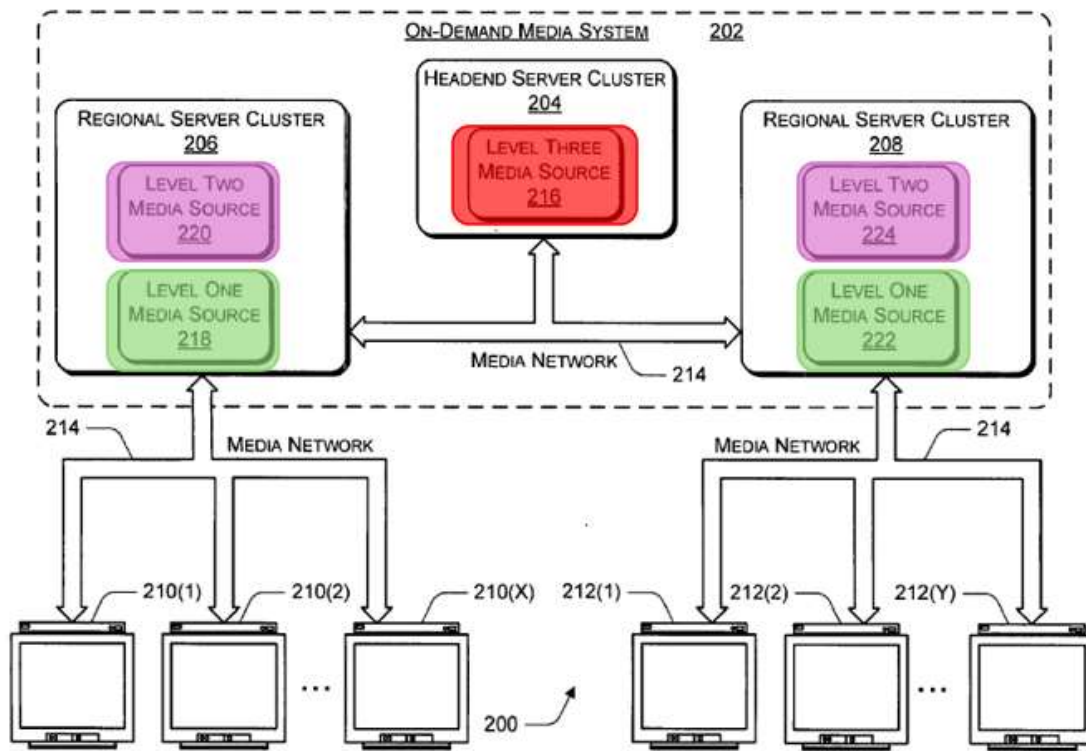


Figure 2

EX-1004, Fig. 2 (annotated); EX-1006, ¶301.

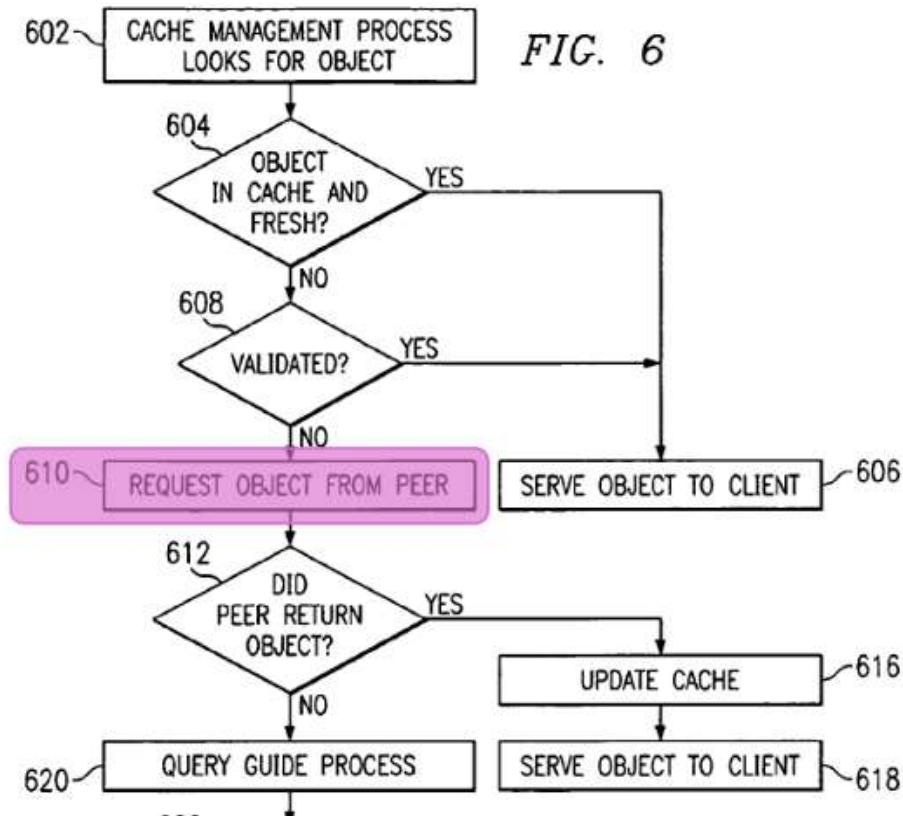
A POSITA would have understood a metro mid-tier server is a commonly used type of server that sits in a metropolis and is thus easily accessible by a large number of end-users. EX-1006, ¶302 (citing EX-1020, 2). A metro server is a logical choice (and is one of a finite number of choices) for this system, as it would serve the greatest number of end-users (located in crowded metropolis areas). By being geographically close to a large number of end-users, the system can leverage the metro servers to reduce latency and efficiently and quickly serve requests from

users located in metropolis areas. EX-1003, 6:7-10; EX-1004, ¶¶[0017], [0020]; EX-1006, ¶302.

In view of the teachings of Dilley and Pai, a POSITA would have been motivated to implement Dilley's parent servers as metro mid-tier servers. EX-1006, ¶303. A POSITA would have recognized such servers as one of a finite set of options, and a POSITA would have recognized their appropriateness for Dilley's system. EX-1006, ¶303; *see also* §VI.B.4.a.

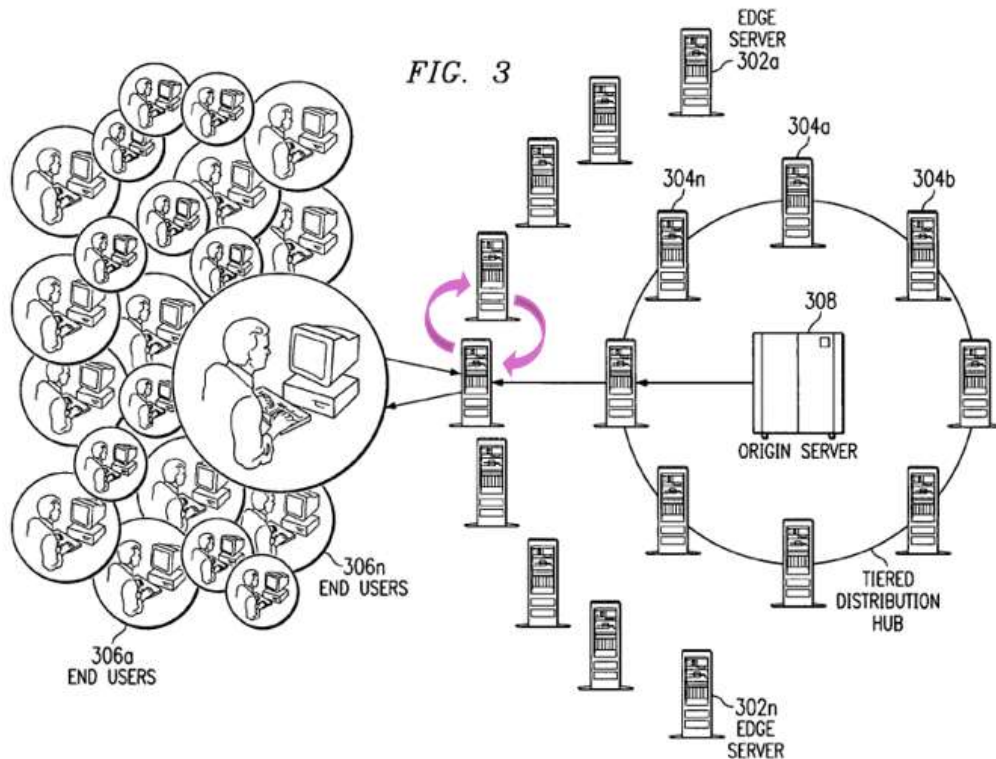
9. Claim 5: “*The method as recited in claim 1 wherein the second server is a peer server of the first tier of servers of the content delivery network.*”

As explained for limitation 1(c), Dilley discloses, in at least one example, that the second server is a peer server of the first tier of servers of the CDN. *Supra* §VI.B.5.d. Dilley explains when a request is received at an edge server, the server first looks for the object in the edge server cache (step 604). EX-1003, 8:5-11. If found, it will serve the object to the client. *Id.* Otherwise, it will request the object from a different server (step 610), and then update the cache to include the requested content. EX-1003, 8:11-18. For instance, Dilley explains that “[i]f the object is not in cache or needs to be validated, the edge server makes a request (e.g., via a cache sharing protocol) to other edge servers in the region (each a ‘peer’) to see if one of them has a fresh copy of the object.” EX-1003, 8:3-15.



EX-1003, Fig. 6 (abbreviated and annotated).

Thus, Dilley discloses that the first server (i.e., the edge server receiving the request from end-user) will request that resource from a second server (i.e., a peer server of the same tier of servers) if the first edge server does not have the requested content in its cache. EX-1003, 5:22-30, 8:11-18; EX-1006, ¶304-306.



EX-1003, Fig. 3 (annotated to identify communication among peer servers).

A POSITA would have further found it obvious for the second server to be a peer server of the first tier of servers because, upon a cache miss at the first server, requesting the content from a peer server in the same tier would reduce latency, ensuring quicker and more reliable content delivery, and would have insulated the parent and origin servers from a large volume of requests (in furtherance of the '573 patent's and Dilley's goals). EX-1003, 1:65-2:3, 6:4-10. EX-1006, ¶307.

Further, as explained above in limitation 1(d), the second server (i.e., the peer server) would send the redirect instruction to the first server when the requested resource is not available at the second server. *Supra* §VI.B.5.e; EX-1006, ¶308.

10. **Claim 6: “*The method as recited in claim 1 wherein the popularity designation associated with the requested resource comprises an indication whether a current popularity value for the resource exceeds a first predetermined popularity threshold.*”**

The combination discloses this limitation for at least the same reasons as for claim 3. *Supra* §§VI.B.4.a, VI.B.7; EX-1006, ¶¶309-311. For instance, Pai will distribute and redistribute media across its three-tiered system based on whether the popularity value of a given content falls below the lowest threshold, exceeds the lowest threshold (but falls below the highest threshold), or exceeds the highest threshold. EX-1004, Fig. 6 (steps 604-612).

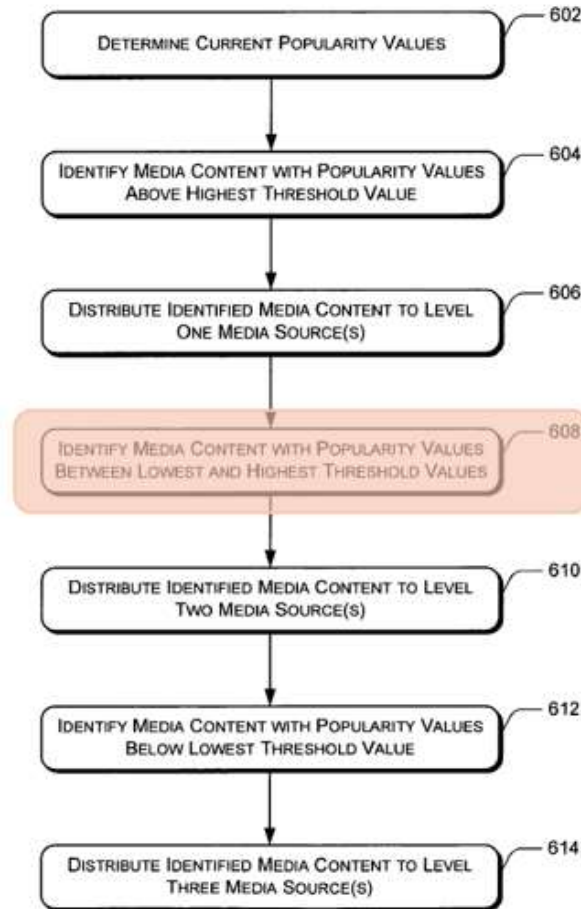


Figure 6

EX-1004, Fig. 6 (annotated).

Specifically, Pai explains that it will “identify media content with popularity values between lowest and highest threshold values” at step 608, shown above. EX-1004, ¶[0042]. A POSITA would have understood that this comprises an indication whether the current popularity value for a resource exceeds a first predetermined popularity threshold (i.e., exceeds the lowest threshold value) and is thus prevented from being classified as the least popular content. EX-1004, Figs. 5-6.

A POSITA would have been motivated to implement Pai's teachings, as described above, with a reasonable expectation of success. *Supra* §VI.B.4.a; EX-1006, ¶¶310-311.

11. Claim 7: “*The method as recited in claim 6 wherein the popularity service is associated with the second server of the content delivery network.*”

The combination teaches that the popularity service is associated with the second server of the CDN because a POSITA would have understood the popularity service could be associated with any server in the CDN. EX-1006, ¶312.

Pai's system is comprised of several servers, each having many components. For instance, Pai explains that its on-demand media server 400 includes exemplary components such as the media distribution module 414, which includes the media scoring module 416 that “determines popularity scores associated with media content.” EX-1005, ¶¶[0025]-[0026]. Thus, each media server 400 can be equipped with a popularity service and, as Pai explains, “on-demand media server 400 may represent a server associated with headend server cluster 204 [in Figure 2], a server associated with one or more regional server clusters 206 and 208 [in Figure 2], or any combination thereof.” EX-1005, ¶[0025]. A POSITA would have understood that the popularity service could be associated with any edge server, parent server, and/or origin server in the CDN. EX-1006, ¶312.

This is also consistent with the teachings of Dilley. Indeed, Dilley explains that in existing systems, each server can be equipped with a hot object cache “for popular objects being served by the CDN.” EX-1003, 4:47-62, Fig. 2. A POSITA would have understood that the hot object cache acts as a popularity service to determine if content is popular and should remain in the server’s cache. EX-1006, ¶313. This is further consistent with the ’573 patent, which explains those of skill in the art would know and understand “that in a multi-tier CDN [(such as those in Dilley, Pai, and Wang)], the popularity service may be located at any tier, or there may be popularity services at more than one tier.” EX-1001, 8:8-11.

In view of the teachings of Dilley and Pai, a POSITA would have understood that the popularity service could be associated with any servers in the CDN. EX-1006, ¶¶314-315. A POSITA would have recognized that associating the popularity service with the second server would have been one of a finite set of options, and a POSITA would have recognized the appropriateness of that structure for Dilley’s system. *Id.*

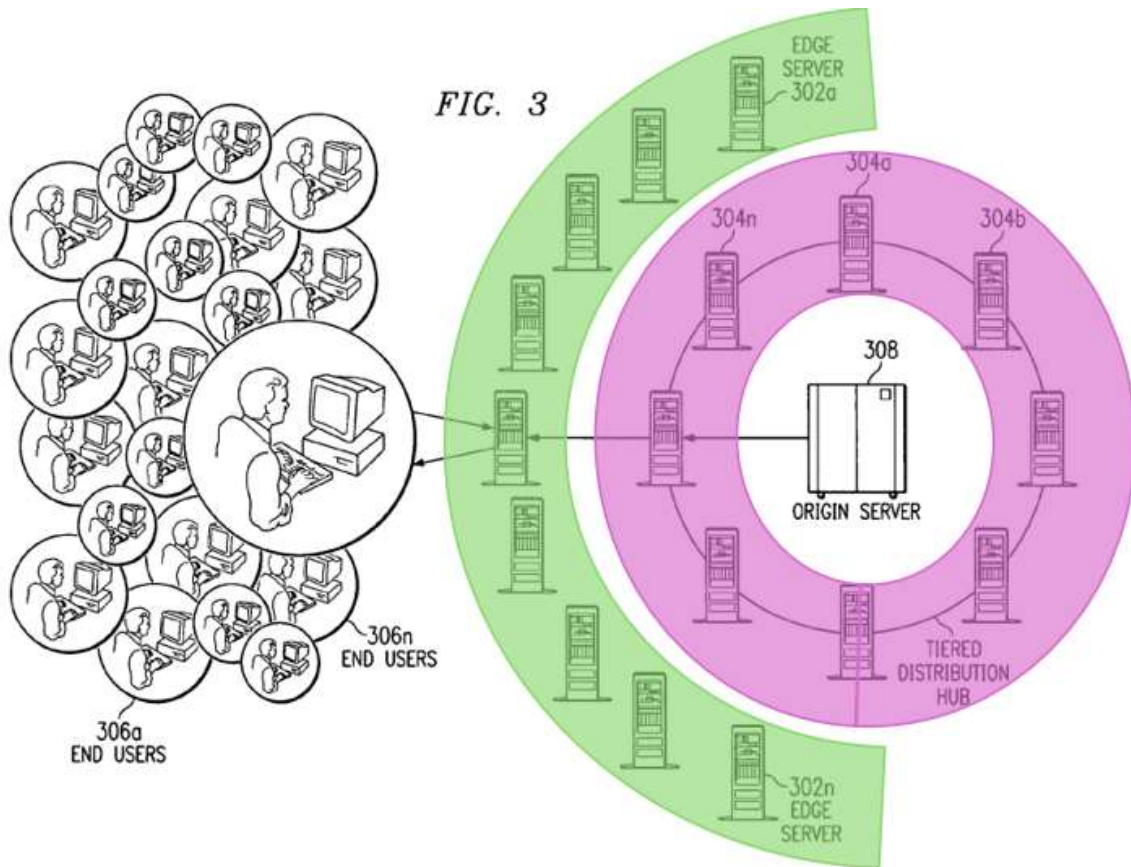
12. Claim 8: “*The method as recited in claim 6 wherein the popularity service is associated with the first server of the first tier of servers of the content delivery network.*”

The combination teaches this limitation because, as explained in §VI.B.11, a POSITA would have understood that the popularity service can be associated with any or all servers in the CDN, including the first server of the first tier of servers

and would have found it obvious for the popularity service to be associated with the first server of the first tier of servers. EX-1006, ¶316.

13. **Claim 9:** “*The method as recited in claim 1 wherein the content delivery network further comprises a second tier of servers distinct from the first tier of servers, and wherein the second server is in the second tier of servers.*”

Dilley discloses this limitation for at least the reasons provided for limitation 1(c). EX-1006, ¶317-318. For instance, Dilley discloses that the parent server is a second server, which is located in a second tier of servers, distinct from the first tier of servers (i.e., the tier encompassing the edge servers).



EX-1003, Fig. 3 (annotated).

Indeed, Dilley employs a tiered distribution system, including, for example, edge servers in one tier, parent servers in a second tier, and an origin server at a third tier. *Id.*; EX-1006, ¶319.

14. Claim 10

- a. *10(a): The method as recited in claim 3 further comprising: receiving, at the first server of the first tier of servers of the content delivery network, an additional request for the resource available from the content delivery network;***

Dilley and Pai seek to identify popular content to serve that content from the edge, improving delivery speeds and insulating origin servers. EX-1003, 6:4-10; EX-1004, ¶[0017]. Thus, a POSITA would have understood the combined system would receive repeated requests for popular content and would go through the same process for each additional request. EX-1003, 8:5-18; EX-1006, ¶320.

For instance, for an initial request, Dilley identifies whether the edge server has the content cached and whether the content is fresh. EX-1003, 8:3-11. If it is not, it will request the item from another server. EX-1003, 8:11-18. When the first server receives this content item, it updates its cache, caching the content at the first server for future requests. *Id.* Thus, when an additional request is received, Dilley's first server will have the object in the cache, and it will be fresh, thus being able to serve the item quickly and reliably to the end-user. EX-1003, Fig. 6; EX-1006, ¶¶321-322.

b. *10(b): determining that the resource is cached at the first server of the first tier of servers; and*

The combination discloses this limitation because one of the initial steps of Dilley's process is to determine whether the resource is cached at the first server. EX-1006, ¶323; EX-1003, 8:3-11, Fig. 6 (at 604). As Dilley explains, if the content is cached and fresh, the first server will serve the content to the end-user. *Id.* Only if the system determines that the content is not cached or is not fresh, will it request the content from another server. EX-1003, 8:11-18. After updating its cache, *supra* §VI.B.7.a, the resource would be available at the first server (within the first tier of servers), and Dilley would accordingly determine the resource was cached there. EX-1006, ¶¶323-324.

c. *10(c): providing the cached resource from the first server of the first tier of servers in response to the additional request for the resource.*

As Dilley explains, once the item is cached at the first server (i.e., the edge server) it will be available for additional requests, such that when Dilley's system receives a request from an end-user, it will check if the requested content is in the cache and is fresh. EX-1003, 8:3-11, 8:15-18. If the content had recently been cached at the edge server from a recent request, *supra* §VI.B.7.a, then it would be cached and fresh and would be served to the end-user from the edge. EX-1003, 8:3-11, Fig. 6 (at 606); EX-1006, ¶¶325-326.

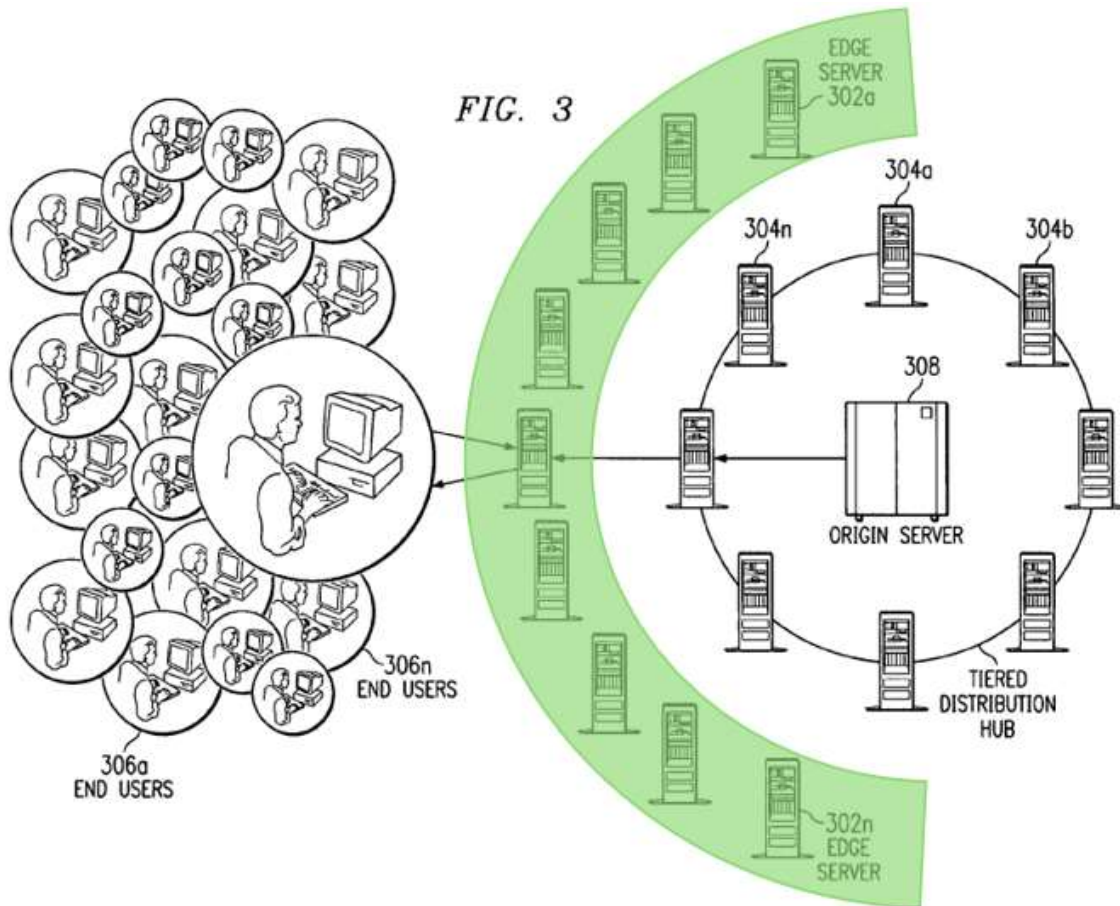
15. Independent Claim 11

a. *11(preamble): A content delivery network comprising:*

The combination discloses this limitation for the same reasons as for limitation 1(preamble). EX-1006, ¶327.

a. *11(a): a first tier of servers comprising a first plurality of servers, an edge server of the first tier of servers receiving a request from a requesting device in communication with the first server for a resource available from the content delivery network;*

The combination discloses this limitation for the same reasons as for limitation 1(a). *Supra* §VI.B.5.b. Specifically, Dilley discloses that the first tier of servers comprises a first plurality of servers:

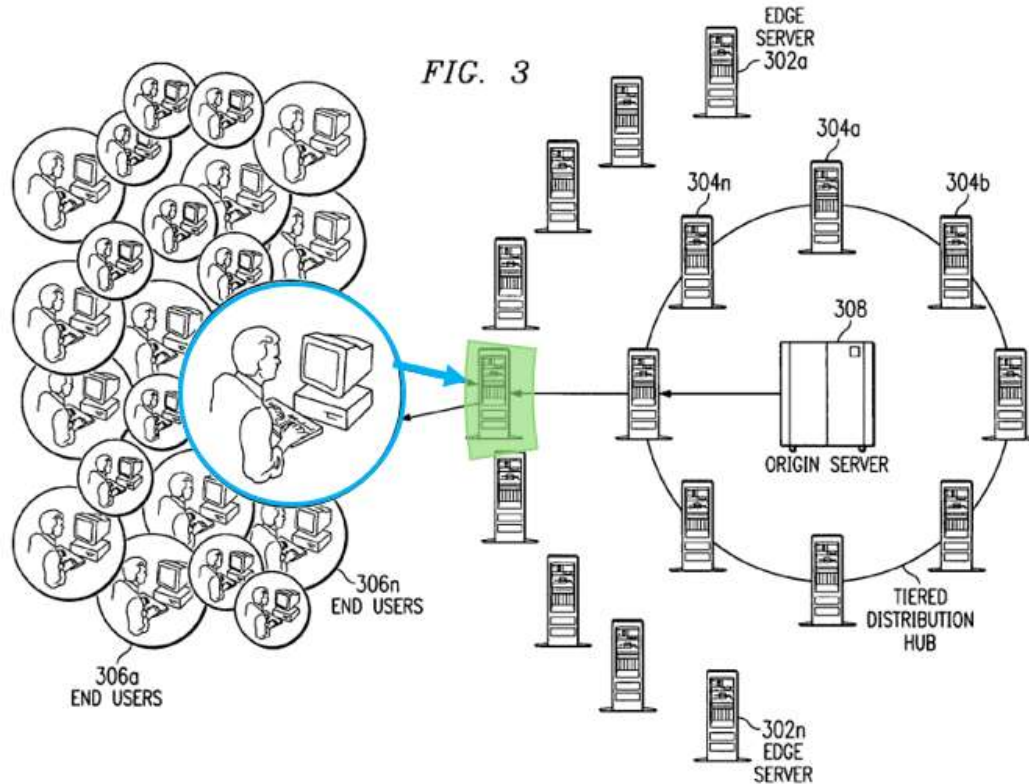


EX-1003, Fig. 3 (annotated).

Indeed, Dilley implements a “tiered distribution” system that uses “two or more tiers: *a set of edge servers and a hierarchical set of ‘parent’ or hub regions that service the edge servers.*” EX-1003, 4:63-5:1 (emphasis added). Dilley discloses that “the CDN service provider operates *a plurality of CDN edge servers 302a-n.*” EX-1003, 5:1-4 (emphasis added), Fig. 3.

As explained, the first server is an edge server of the first tier of servers and is in communication with the requesting device because it is the edge server that

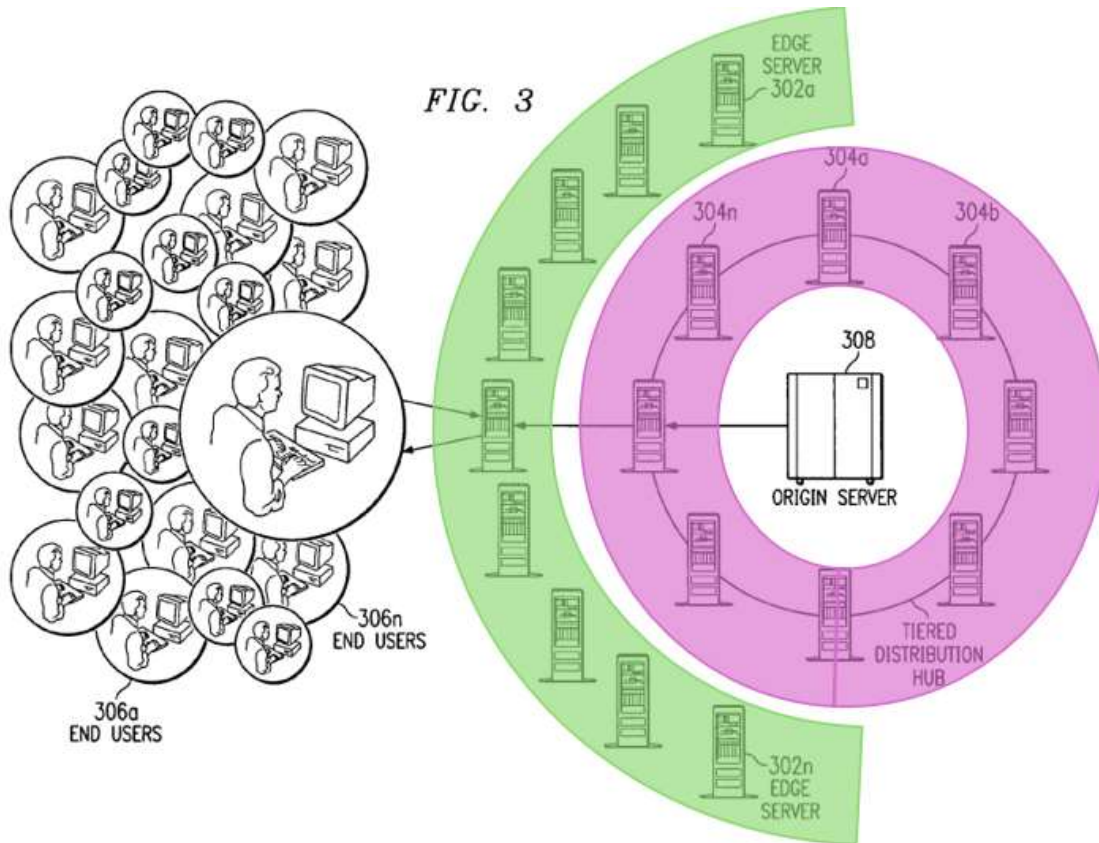
“receives end-user requests for origin server content and serves this content.” EX-1003, 7:50-53.



See, e.g., EX-1003, Fig. 3 (annotated) (showing edge server receiving request from end user); EX-1006, ¶328-331.

b. 11(b(i)): a second tier of servers comprising a second plurality of servers,

The combination discloses this limitation for the same reasons as for limitation 1(c) and claim 9. *Supra* §§VI.B.5.d, VI.B.13. More specifically, Dilley discloses that the second tier of servers comprises a second plurality of servers:



EX-1003, Fig. 3 (annotated).

As explained, Dilley implements a “tiered distribution” system that uses “two or more tiers: a set of edge servers and a hierarchical *set* of ‘parent’ or hub regions that service the edge servers.” EX-1003, 4:63-5:1 (emphasis added). Dilley further discloses that “the CDN service provider operates a plurality of CDN edge servers 302a-n and a set of tiered distribution hubs 304a-n.” EX-1003, 5:1-4. These tiered distribution hubs encompass a second plurality of servers within a second tier of servers and are used to “buffer the origin server 308” and “ensure that the edge servers have the information requested by the end users.” EX-1003, 5:4-10; EX-1006, ¶¶332-334.

- c. ***11(b(ii)): a first server of the second plurality of servers receiving a request from the edge server for the content and, in response, transmitting a redirect instruction intended for the requesting device to the edge server to obtain the resource from a content server and transmitting an instruction to not cache the portion of the resource at the edge server when the portion of the resource is obtained from the content server; and***

The combination discloses this limitation for the same reasons as for limitations 1(c)-1(e). *Supra* §§VI.B.5.d-f. Moreover, a POSITA would have also understood that the instruction to not cache the portion of the resource at the edge server could be transmitted from the first server of the second plurality of servers (i.e., the parent server), and that doing so would have been one of a finite number of design choices. EX-1006, ¶335. As in the discussion for limitation 1(d), having the parent server transmit the instruction to not cache would similarly be useful to free the content server for other tasks. *Id.*

- d. ***11(c): a popularity service tracking a popularity designation associated with the requested resource;***

The combination discloses this limitation for the same reasons as for limitation 1(b). *Supra* §VI.B.5.c; EX-1006, ¶336.

- e. ***11(d): wherein the edge server processes the redirect instruction to obtain the content from the content server and provides the obtained content to the requesting device.***

The combination discloses this limitation for the same reasons as for limitations 1(d)-1(f). *Supra* §§VI.B.5.e-g; EX-1006, ¶337.

16. Claims 12-18

The combination discloses these claims for at least the same reasons as for claims 2-8, respectively. *Supra* §§VI.B.6-12; EX-1006, ¶¶338-347. Moreover, for claim 13, a POSITA would have understood that the instruction to cache the portion of the resource at the edge server could be transmitted from the first server of the second plurality of servers (i.e., parent server), and that doing so would have merely been one of a finite number of design choices. EX-1006, ¶342. As in the discussion for limitation 1(d), having the parent server transmit the instruction would similarly be useful to free the content server for other tasks. *Id.*

17. Independent Claim 19

- a. ***19(preamble)-19(c): A method of content delivery in a content delivery network comprising: receiving, at a first server of a first tier of servers of the content delivery network, a request from a requesting device for a resource available from the content delivery network; accessing a popularity service associated with the first server of the first tier of the content delivery network to determine a popularity designation associated with the requested resource; receiving an instruction to not cache the resource at the first server of the first tier of servers when the portion of the resource is obtained from the content server of the content delivery network;***

The combination discloses these limitations for the same reasons as for limitations 1(preamble), 1(a), 1(b), and 1(e), respectively, and, the combination also discloses limitation 19(b) for the additional reasons provided above for claim 8. *Supra* §§VI.B.5.a-f, VI.B.12; EX-1006, ¶¶348-351.

- b. ***19(d): requesting the resource from a second server of the content delivery network based at least on the popularity designation associated with the requested resource; and***

The combination discloses this limitation for the same reasons as for limitation 1(c). *Supra* §VI.B.5.d. Furthermore, Dilley explains that “the servers in the tiered distribution hubs [(i.e., the parent or origin servers)] will provide the edge servers with the requested content” when the “edge servers do not have the appropriate file when it is requested.” EX-1003, 5:6-10. Because the edge servers are used to prevent the origin servers from becoming “overwhelmed with requests for popular

content,” a POSITA would have understood that content designated as unpopular would be served from the parent or origin servers, while popular content would be served from the edge. EX-1003, 5:10-21. Thus, based on an unpopular designation, Dilley would retrieve content from the second server. EX-1006, ¶¶352-353.

Further, Pai dynamically manages and distributes content based on popularity values. EX-1004, ¶[0013]. As Pai explains, as content is consumed, its popularity value can change over time, moving from media source to media source. EX-1004, ¶¶[0020], [0026], [0029]. Pai explains that media with popularity values exceeding the highest threshold will be distributed and managed in one tier—media source one—while others would be distributed and managed in other tiers of media sources. EX-1004, ¶¶[0038]-[0045]. Thus, a POSITA would have understood that, when Pai’s system receives a request for content that does not exceed the highest popularity threshold, the request would need to be forwarded to a second server (i.e., a server in level two for mid-tier popular content or a server in level three for unpopular content). EX-1006, ¶354.

As provided above, a POSITA would have thus implemented Pai’s disclosure of dynamically redistributing content based on its popularity into Dilly’s popularity service, thereby distributing the “flash-popular content” at the edge “closer to the requesting end-users,” EX-1003, 9:57-65, the mid-level popular items at the middle

tier servers, and the least popular items at the origin furthest from the requesting end-user. *Supra* §VI.B.4.a; EX-1006, ¶355.

c. *19(e): providing the obtained resource to the requesting device.*

The combination discloses this limitation for the same reasons as for limitation 1(f). *Supra* §VI.B.5.g; EX-1006, ¶356.

18. Claim 20: “*The method as recited in claim 19 wherein the second server of the content delivery network is a content server storing the requested content.*”

The combination discloses this limitation because Dilley discloses a “tiered infrastructure using two or more tiers: a set of edge servers and a hierarchical set of ‘parent’ or hub regions that service the edge servers.” EX-1003, 4:63-5:10. Thus, a POSITA would have understood that in a two-tiered system, the first tier would encompass edge servers, and the second would comprise the higher-level servers, performing the task of the parent and origin servers. EX-1006, ¶357. Thus, the second server in the two-tiered system would likewise be content servers storing the content and “servic[ing] the edge servers.” EX-1003, 4:64-5:10. Indeed, Dilley explains the edge servers “include[] a cache for storing content provider content,” and are thus content servers. EX-1003, 7:50-53. The parent and origin servers also store content and are similarly content servers. EX-1006, ¶357. Dilley explains that when an edge server does not have a content, it will reach out to a parent server or content server, which will serve the content from their cache, indicating that they

are also content servers. EX-1003, 8:53-58. The two-tiered system would still achieve Dilley’s goals of quickly and reliably serving content, while having the edge servers insulate the content server from repeated requests for popular content. EX-1003, 2:12-17, 2:26-31, 5:10-21; EX-1006, ¶357.

VII. THE BOARD SHOULD NOT EXERCISE DISCRETION TO DENY INSTITUTION

A. 35 U.S.C. §314(a) Does Not Favor Denial

The *Fintiv* factors weigh strongly in favor of institution.

Factor 1 is neutral because, although Petitioner will move to stay the litigation, (*see* §VII.B), the district court has not considered or ruled on any stay motion. *Sotera Wireless, Inc. v. Masimo Corp.*, IPR2020-01019, Paper 12 at 14 (P.T.A.B. Dec. 1, 2020) (precedential).

Factor 2 is neutral. The trial is presently scheduled for September 28, 2026, which aligns closely with the Board’s anticipated Final Written Decision in this case. EX-1015, 3. Assuming Patent Owner files a preliminary response three months from this Petition (August 2025) and the Board institutes review three months later (November 2025), the Board’s anticipated Final Written Decision will be November 2026—about a month from the current trial date.

Factor 3 favors institution because the Court and the parties have expended few resources in litigation. Fact discovery opened in March 2025, and the Markman hearing is not until September 12, 2025. *Id.*, 2. Petitioner promptly brought this

challenge approximately three months after receiving infringement contentions (and notice of the asserted claims) on January 14, 2025. *CoolIT Sys., Inc. v. Asetek Danmark A/S*, IPR2021-01195, Paper 10 at 11-14 (P.T.A.B. Dec. 28, 2021); *Sotera*, Paper 12 at 16-17 (petition filed two months after infringement contentions and two weeks before the statutory deadline deemed reasonable).

Factor 4 favors institution. This Petition challenges twenty claims, which will be greater than the number of claims tried (which will be no more than seven based on the infringement contentions). Ex-1016-1017. Resolving invalidity questions here would mitigate duplicative efforts in this litigation and in any future cases. Petitioner also stipulates that, if this Petition is instituted, it will not pursue in district court the grounds identified in this Petition or that could have been reasonably raised in the IPR (i.e., any ground that reasonably could have been raised under 35 U.S.C. §§102 or 103 on the basis of prior art patents or printed publications) with respect to the claims subject to institution. Ex-1018; *Sotera*, Paper 12 at 18-19 (finding this “mitigates any concerns of duplicative efforts between the district court and the Board, as well as concerns of potentially conflicting decisions.”).

Under Factor 5, Petitioner is the defendant in the litigation.

Factor 6 further favors institution. To Petitioner’s knowledge, no other party has sought the Office’s review of the ’573 patent, and indeed, Petitioner is aware of no other party likely to do so, minimizing any likelihood of serial or parallel

petitions. Petitioner relies on prior art that the Office never applied, presents significantly different invalidity grounds, and relies on Dr. Mowry's declaration explaining each claim's invalidity. The significant public interest against "leaving bad patents enforceable" further supports institution under this Factor. *Thryv, Inc v. Click-To-Call Techs., LP*, 140 S. Ct. 1367, 1374 (2020).

B. 35 U.S.C. §325(d) Does Not Favor Denial

First, there is no evidence Examiner evaluated the Applicant's priority claim or any prior art under a proper, later February 25, 2019, priority date, as presented here in Ground 1.

Second, the factors in *Becton, Dickinson & Co. v. B. Braun Melsungen AG*, IPR2017-01586, Paper 8 at 17-18 (P.T.A.B. Dec. 15, 2017) favor institution. *See also Advanced Bionics, LLC v. MED-EL Elektromedizinische Geräte GmbH*, IPR2019-01469, Paper 6 at 8-11 (P.T.A.B. Feb. 13, 2020) (precedential).

Advanced Bionics, step 1, and *Becton, Dickinson* factors (a), (b), and (d) favor institution because Pai and Wang were not identified during prosecution. While Dilley appears on the face of the patent, it is listed as one of over 500+ other references, and there is no evidence that the Examiner considered any particular disclosures in Dilley relative to the claims of the '573 patent. *See, e.g., Microsoft Corp. v. Parallel Networks Licensing LLC*, IPR2015-00486, Paper 10, 15–20 (P.T.A.B. July 15, 2015) (granting institution when "the references were not applied

against the claims and there [was] no evidence that the Examiner considered the particular disclosures cited” in an IDS). While a similar Dilley reference (not the Dilley submitted with this Petition) was considered during examination of an earlier priority application (U.S. Patent Application No. 12/408,681), Petitioner relies on Dilley for features that the Examiner agreed were taught by the prior art and that Patent Owner never disputed. EX-1021, 2230-2240, 2254-2258; EX-1022. In any event, Dilley was never considered with the additional references cited in Ground 2 (Pai and Wang), which were not before the Office during prosecution. Further, the Examiner did not have the benefit of Dr. Mowry’s expert declaration during prosecution. The references are not cumulative of the prosecution prior art because they teach and renders obvious all challenged claims.

VIII. MANDATORY NOTICES UNDER 37 C.F.R. §42.8

A. Real Parties-in-Interest

The Petitioner and real party-in-interest is Google LLC.⁶

B. Related Matters

The ’573 patent is, or has been, involved in the following proceedings:

⁶ 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.

Name	Number	Forum	Filed
<i>Sandpiper CDN, LLC v. Google LLC</i>	2-24-cv-03951	C.D. Cal.	May 10, 2024

C. Lead and Back-Up Counsel, and Service Information

Petitioner provides the following designation of counsel:

Lead Counsel	Back-Up Counsel
Erika H. Arner (Reg. No. 57,540) erika.arner@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 1875 Explorer Street Suite 800 Reston, VA 20190-6023 Tel: 571-203-2700; Fax: 202-408-4400	Daniel C. Tucker (Reg. No. 62,781) daniel.tucker@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 1875 Explorer Street, Suite 800 Reston, VA 20190-6023 Tel: 571-203-2700 Fax: 202-408-4400 Kara A. Specht (Reg. No. 69,560) kara.specht@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 271 17th Street, NW Suite 1400 Atlanta, GA 30363-6209 Tel: 571-203-2700 Fax: 404-653-6444 Cara R. Regan (Reg. No. 70,209) cara.regan@finnegan.com Sydney R. Kestle (Reg. No. 78,725) sydney.kestle@finnegan.com Anthony J. Berlenbach (Reg. No. 77,963) anthony.berlenbach@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 901 New York Avenue, NW

	Washington, DC 20001-4413 Tel: 202-408-6013 Fax: 202-408-4400
--	---

In a concurrently filed Power of Attorney, Google LLC has granted Power of Attorney to Practitioners at Finnegan, Henderson, Farabow, Garrett & Dunner, LLP.

Petitioner consents to service by email at the addresses listed above and Google-Sandpiper-IPRs@finnegan.com.

IX. CONCLUSION

Petitioner respectfully requests the Board grant IPR and find all challenged claims unpatentable.

Respectfully submitted,

Dated: May 1, 2025

By: /Erika H. Arner/
Erika H. Arner (Reg. No. 57,540)

CERTIFICATION UNDER 37 C.F.R. §42.24(D)

Pursuant to 37 C.F.R. §42.24(d), the undersigned hereby certifies that the foregoing Petition contains 13,909 words, excluding those portions identified in 37 C.F.R. §42.24(a), as measured by the word-processing system used to prepare this paper.

Respectfully submitted,

Dated: May 1, 2025

By: /Erika H. Arner/
Erika H. Arner (Reg. No. 57,540)

CERTIFICATE OF SERVICE

The undersigned certifies that, in accordance with 37 C.F.R. §§42.6(e) and 42.105, the **Petition for *Inter Partes* Review of U.S. Patent No. 10,924,573, Power of Attorney, and all associated Exhibits** were served on May 1, 2025 via FedEx Priority Overnight® on the correspondence address of record indicated in the Patent Office's Patent Center system for U.S. Patent No. 10,924,573:

ATTN: Patent Docketing
Level 3 Communications, LLC
931 14th St.
Denver, CO 80202

Dated: May 1, 2025

By: /Lisa C. Hines/
Lisa C. Hines
Case Manager
Finnegan, Henderson, Farabow,
Garrett & Dunner, LLP