

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MICROSOFT CORPORATION,
Petitioner

v.

SANDPIPER CDN, LLC,
Patent Owner.

PTAB Case No. IPR2026-00190

U.S. Patent No. 9,456,053 B2

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 9,456,053 B2**

TABLE OF CONTENTS

[illegible]

13.	Claims 14/40	30
14.	Claims 15/41	31
15.	Claims 16/42	31
16.	Claim 21	32
17.	Claims 22/47(pre)	33
18.	Claims 23/47(E)-(F).....	33
19.	Claim 49	34
20.	Claims 25/50	34
B.	Ground 2A: Obviousness over Liu and Devanneaux (Claims 1-2, 4, 6, 9-16, 18-28, 30, 32, 35-42, 44-50)	35
1.	Prior Art	35
2.	Motivation to Combine Liu with Devanneaux	38
3.	Claims 1, 27	41
4.	Claims 2/28	49
5.	Claims 4/30	50
6.	Claims 6/32	51
7.	Claims 9/35	52
8.	Claims 10/36	53
9.	Claims 11/37	54
10.	Claims 12/38	55
11.	Claims 13/39	57
12.	Claims 14/40	58
13.	Claims 15/41	58
14.	Claims 16/42	58
15.	Claims 18/44	59
16.	Claims 19/45	60
17.	Claims 20/46	61
18.	Claim 21	62
19.	Claims 22/47(pre)	62

20.	Claims 23/47(E)-(F).....	63
21.	Claims 24/48	64
22.	Claim 49	66
23.	Claims 25/50	66
24.	Claim 26.....	66
C.	Ground 2B: Obviousness over Liu, Devanneaux, and Nygren (Claims 17, 43).....	67
1.	Prior Art	67
2.	Claims 17/43	68
D.	Ground 2C: Obviousness over Liu, Devanneaux, and Lewin819 (Claims 3, 5, 7-8, 29, 31, 33-34).....	69
1.	Prior Art	69
2.	Motivation to Combine with Lewin819.....	71
3.	Claims 3/29	73
4.	Claims 5/31	74
5.	Claims 7/33	77
6.	Claims 8/34	77
E.	Ground 2D: Obviousness over Liu, Devanneaux, and Crosbie (Claims 18-20, 44-46).....	77
1.	Prior Art	77
2.	Motivation to Combine with Crosbie	78
3.	Claims 18/44	80
4.	Claims 19/45	81
5.	Claims 20/46	82
VIII.	CONCLUSION.....	83
IX.	MANDATORY NOTICES	83
A.	Real Party in Interest.....	83
B.	Related Matters.....	83
C.	Lead and Back-Up Counsel, and Service Information	83

D. Power of Attorney	84
----------------------------	----

EXHIBIT LIST

No.	Exhibit
1001	U.S. Patent No. 9,456,053 (“’053 patent”)
1002	File History of U.S. Application 13/714,410 (“’053 File History”)
1003	Declaration of Mr. Mark Lanning
1004	Curriculum Vitae of Mr. Mark Lanning
1005	U.S. Patent 7,010,578 (“Lewin”)
1006	U.S. Patent 7,240,100 (“Wein”)
1007	U.S. Publication 2010/0257258 (“Liu”)
1008	U.S. Publication 2007/0156845 (“Devanneaux”)
1009	Erik Nygren, et al., <i>The Akamai Network: A Platform for High-Performance Internet Applications</i> , 2010 (“Nygren”)
1010	U.S. Publication 20100274819 (“Lewin819”)
1011	International Application Publication WO 001/13456 (“Crosbie”)
1012	U.S. Publication 2002/0065899 (“Smith”)
1013	U.S. Publication 2011/0082982 (“Harvell”)
1014	U.S. Patent 8,180,720 (“Kovacs”)
1015	U.S. Publication 2013/0103556 (“Lyon”)
1016	U.S. Patent 7,127,713 (“Davis”)
1017	Jennifer Mears, <i>Buyout ends Akamai, Speedera feud</i> (“Mears”), March 21, 2005 (“Mears”)
1018	U.S. Patent 6,343,236 (“Gibson”)

No.	Exhibit
1019	U.S. Publication 2002/0129363A1 (“ <i>McGuire</i> ”)
1020	<i>HTTP, The Definitive Guide</i> by David Gourley, et al. (“ <i>Gourley</i> ”)
1021	<i>Globally Distributed Content Delivery</i> by John Dilley, et al. (“ <i>Dilley</i> ”)
1022	<i>Content Delivery Networks</i> by Rajkumar Buyya, et al. (“ <i>Buyya</i> ”)
1023	U.S. Patent 7,024,466 (“ <i>Outten</i> ”)
1024	<i>Caching and Prefetching for Web Content Distribution</i> by Jianliang Xu, et al. (“ <i>Xu</i> ”)
1025	U.S. Publication 2006/0015574 (“ <i>Seed</i> ”)
1026	U.S. Publication 2002/0004824 (“ <i>Cuan</i> ”)
1027	U.S. Publication 2001/0017707 (“ <i>Lee</i> ”)
1028	U.S. Publication 2007/0118617 (“ <i>Lee617</i> ”)
1029	U.S. Patent 7,340,510 (“ <i>Liskov</i> ”)
1030	U.S. Publication 2004/0221019 (“ <i>Swildens</i> ”)
1031	U.S. Publication 2004/0133629 (“ <i>Reynolds</i> ”)
1032	U.S. Publication 2006/0230104 (“ <i>Winter</i> ”)
1033	Declaration of June Munford (“ <i>Munford Declaration</i> ”)
1034	<i>Sandpiper CDN, LLC v. Microsoft Corp.</i> , No. 2:25-cv-00664 (E.D. Tex.), Docket

LISTING OF CHALLENGED CLAIMS

U.S. Patent No. 9,456,053 B2
Claim 1 (Cl.1)
1(pre) A content delivery network (CDN), the CDN serving resources on behalf of one or more customers of the CDN, the CDN comprising:
1(a) a control core; and
1(b) a plurality of cache servers, each particular cache server of said plurality of cache servers being constructed and adapted to:
1(b)(1) upon joining the CDN,
1(b)(1)(1) obtain global configuration data from the control core; and
1(b)(2) having joined the CDN,
1(b)(2)(1) obtain updated global configuration data, if needed; and
1(b)(2)(2) obtain customer configuration information associated with at least one customer of the CDN;
1(b)(2)(3) serve to a client a particular resource associated with a particular customer of the CDN in accordance with (i) the global configuration data, and (ii) customer configuration information obtained in (b)(2)(2) and associated with the particular customer.
Claim 2 (Cl.2)
2. The CDN of claim 1 wherein each particular cache server of said plurality of cache servers is constructed and adapted to, upon joining the CDN, (b)(1)(2) obtain data from one or more other locations in the CDN.
Claim 3 (Cl.3)
3. The CDN of claim 1 wherein processing by a cache server in (b)(2)(2) to obtain customer configuration information associated with a specific customer occurs upon receipt of a request for a resource associated with the specific customer.
Claim 4 (Cl.4)

4. The CDN of claim 1 wherein processing by a cache server in (b)(2)(2) to obtain customer configuration information associated with a specific customer occurs prior to receipt of a request for a resource associated with the specific customer.
Claim 5 (Cl.5)
5. The CDN of claim 1 wherein customer configuration information associated with the particular customer comprises one or more scripts to be used by the particular cache server to process requests for resources associated with the particular customer.
Claim 6 (Cl.6)
6. The CDN of claim 1 wherein the customer configuration information comprises at least one CDN resource.
Claim 7 (Cl.7)
7. The CDN of claim 1 wherein the customer configuration information associated with the particular customer comprises at least one customer configuration script (CCS) associated with the particular customer.
Claim 8 (Cl.8)
8. The CDN of claim 1 wherein the customer configuration information associated with the particular customer specifies customer-specific processing requirements for resources served on behalf of that customer.
Claim 9 (Cl.9)
9. The CDN of claim 1 wherein the global configuration data comprises at least one CDN resource.
Claim 10 (Cl.10)
10. The CDN of claim 9 wherein the global configuration data comprises global configuration object (GCO).
Claim 11 (Cl.11)
11. The CDN of claim 2 wherein the one or more other locations in the CDN comprise at least one location selected from: a cache server, a cache server site, a region of cache servers, a cache cluster, and a cache cluster site.
Claim 12 (Cl.12)

12. The CDN of claim 11 wherein the one or more other locations in the CDN comprise at least one peer location.
Claim 13 (Cl.13)
13. The CDN of claim 2 wherein the particular cache server determines said one or more other locations using said information in said global configuration data.
Claim 14 (Cl.14)
14. The CDN of claim 1 wherein said particular cache server obtains said updated global configuration data from one or more locations in the CDN.
Claim 15 (Cl.15)
15. The CDN of claim 14 wherein the one or more locations in the CDN comprise one or more locations selected from: (i) the control core; and (ii) one or more locations selected from: a cache server, a cache server site, a region of cache servers, a cache cluster, and a cache cluster site.
Claim 16 (Cl.16)
16. The CDN of claim 1 wherein the control core comprises a distributed system consisting of a plurality of machines.
Claim 17 (Cl.17)
17. The CDN of claim 16 wherein the control core uses a distributed consensus algorithm to achieve consensus among the plurality of machines.
Claim 18 (Cl.18)
18. The CDN of claim 1 wherein each particular cache server of said plurality of cache servers is further constructed and adapted to: determine validity of a version of global configuration data stored on said particular cache server, and wherein said particular cache server obtains updated global configuration data in (b)(2)(1) when said particular cache server determines that said version of said global configuration data stored on said particular cache server is invalid.
Claim 19 (Cl.19)
19. The CDN of claim 1 wherein each particular cache server of said plurality of cache servers is further constructed and adapted to: determine validity of a version of particular customer configuration information associated with a particular customer and stored on said particular cache server, and wherein said particular cache server obtains customer configuration information in (b)(2)(2)

when said particular cache server determines that said version of said particular customer configuration information stored on said particular cache server is invalid.

Claim 20 (Cl.20)

20. The CDN of claim 19 wherein said particular cache server is constructed and adapted to determine said validity of said version of particular customer configuration information stored on said particular cache server in response to a request to serve content associated with said particular customer.

Claim 21 (Cl.21)

21. The CDN of claim 1 further comprising: (c) at least one rendezvous mechanism.

Claim 22 (Cl.22)

22. The CDN of claim 1 further comprising: (d) a collector system comprising one or more collector mechanisms, each of said collector mechanisms being constructed and adapted to: (d)(1) obtain information from at least one cache server.

Claim 23 (Cl.23)

23. The CDN of claim 22 wherein each particular cache server of said plurality of cache servers is further constructed and adapted to: (b)(3) generate information; and (b)(4) provide at least some of said generated information to said collector system.

Claim 24 (Cl.24)

24. The CDN of claim 21 wherein each particular cache server of said plurality of cache servers provides at least some of said generated information in (b)(4) by streaming said at least some of said generated information to one or more collector mechanisms in said collector system.

Claim 25 (Cl.25)

25. The CDN of claim 1 wherein the CDN has customer configuration information associated therewith, and wherein each particular cache server serves CDN resources in accordance with the global configuration data and the customer configuration information associated with the CDN.

Claim 26 (Cl.26)

26. The CDN of claim 1 wherein at least one cache server of said plurality of cache servers comprises an executive system supporting concurrent processing of tasks on said at least one cache server.
Claim 27 (Cl.27)
27(pre) A computer-implemented method, operable in a content delivery network (CDN) comprising:
27(pre)(a) a control core; and
27(pre)(b) a plurality of cache servers, the CDN serving resources on behalf of one or more customers of the CDN, the method comprising, by hardware and software on a particular cache server of said plurality of cache servers:
27(A) obtaining global configuration data from the control core;
27(B) selectively obtaining updated global configuration data; and
27(C) obtaining customer configuration information associated with a particular customer of the CDN;
27(D) serving, to a client, a particular resource associated with the particular customer in accordance with (i) the global configuration data; and (ii) the customer configuration information associated with the particular customer.
Claim 28 (Cl.28)
28. The method of claim 27 further comprising: (A)(2) obtaining data from one or more other locations in the CDN.
Claim 29 (Cl.29)
29. The method of claim 27 wherein the obtaining of said customer configuration information in (C) occurs upon receipt of a request of said particular cache server for a resource associated with the particular customer.
Claim 30 (Cl.30)
30. The method of claim 27 wherein the obtaining of said customer configuration information in (C) occurs prior to receipt of a request of said particular cache server for a resource associated with the particular customer.
Claim 31 (Cl.31)

31. The method of claim 27 wherein the customer configuration information comprises one or more scripts to be used by the particular cache server to process requests for resources associated with the particular customer.
Claim 32 (Cl.32)
32. The method of claim 27 wherein the customer configuration information comprises at least one CDN resource.
Claim 33 (Cl.33)
33. The method of claim 27 wherein the customer configuration information comprises at least one customer configuration script (CCS) associated with the particular customer.
Claim 34 (Cl.34)
34. The method of claim 27 wherein the customer configuration information associated with the particular customer specifies customer-specific processing for resources served on behalf of that customer.
Claim 35 (Cl.35)
35. The method of claim 27 wherein the global configuration data comprises at least one CDN resource.
Claim 36 (Cl.36)
36. The method of claim 35 wherein the global configuration data comprises global configuration object (GCO).
Claim 37 (Cl.37)
37. The method of claim 28 wherein the one or more other locations in the CDN comprise at least one location selected from: a cache server, a cache server site, a region of cache servers, a cache cluster, and a cache cluster site.
Claim 38 (Cl.38)
38. The method of claim 37 wherein the one or more other locations in the CDN comprise at least one peer location.
Claim 39 (Cl.39)

39. The method of claim 28 wherein the particular cache server determines said one or more other locations using said information in said global configuration data.
Claim 40 (Cl.40)
40. The method of claim 27 wherein said particular cache server obtains said updated global configuration data in (B) from one or more locations in the CDN.
Claim 41 (Cl.41)
41. The method of claim 40 wherein the one or more locations in the CDN comprise one or more locations selected from: (i) the control core; and (ii) one or more locations selected from: a cache server, a cache server site, a region of cache servers, a cache cluster, and a cache cluster site.
Claim 42 (Cl.42)
42. The method of claim 27 wherein the control core comprises a distributed system consisting of a plurality of machines.
Claim 43 (Cl.43)
43. The method of claim 42 wherein the control core uses a distributed consensus algorithm to achieve needed consensus among the plurality of machines.
Claim 44 (Cl.44)
44. The method of claim 27 further comprising, by said particular cache server: (F) determining validity of a version of global configuration data stored on said particular cache server, and wherein said particular cache server obtains updated global configuration data in (B) when said particular cache server determines that said version of said global configuration data stored on said particular cache server is invalid.
Claim 45 (Cl.45)
45. The method of claim 27 further comprising, by said particular cache server: (G) determining validity of a version of particular customer configuration information associated with the particular customer and stored on said particular cache server, and wherein said particular cache server obtains customer configuration information in (C) when said particular cache server determines that said version of said particular customer configuration information stored on said particular cache server is invalid.

Claim 46 (Cl.46)
46. The method of claim 45 wherein said particular cache server determines validity of said version of particular customer configuration information in (G) in response to a request to serve content associated with said particular customer.
Claim 47 (Cl.47)
47(pre) The method of claim 27 wherein the CDN further comprises: (d) a collector system comprising one or more collector mechanisms, each of said collector mechanisms being constructed and adapted to obtain information from at least one cache server, the method further comprising, by said particular cache server:
47(E) generating information; and
47(F) providing at least some of said generated information to said collector system.
Claim 48 (Cl.48)
48. The method of claim 47 wherein said providing in (F) comprises: (F)(1) streaming said at least some of said generated information to one or more collector mechanisms in said collector system.
Claim 49 (Cl.49)
49(pre) The method of claim 27 further comprising, by said particular cache server of said plurality of cache servers:
49(E) obtaining second customer configuration information associated with a second customer of the CDN, said second customer being distinct from said particular customer; and
49(F) serving a second particular resource associated with the second customer in accordance with (i) the global configuration data, and (ii) the second customer configuration information associated with the second customer.
Claim 50 (Cl.50)
50. The method of claim 27 wherein the CDN has customer configuration information associated therewith, and wherein the particular cache server serves CDN resources in accordance with the global configuration data and the customer configuration information associated with the CDN.

By 2011, many third-party companies had built “content-delivery networks” (“CDNs”) to relieve content owners of the technical burdens of serving their content—such as websites, music, movies, and more—over the Internet. CDNs were networks of geographically distributed servers that stored (“cached”) and served content to end users on behalf of content owners to avoid overloading the content owner’s servers (like how Amazon stores and ships products from its own warehouses on behalf of other sellers, so sellers do not need their own warehouses and distribution). As such, CDN operators needed to be able to configure their networks as they saw fit, while also providing content owners some control over the delivery of their content.

U.S. Patent No. 9,456,053 (Ex-1001) did nothing but lay claim to the known need that both CDN operators and content owners have some say in how CDN networks deliver content to end users. Servers would receive some (“global”) configurations from the CDN operator to control the network, and other (“customer”) configurations from the content owners that controlled delivery of the customer’s specific content on the network. Content would then be delivered pursuant to both sets of configurations. That idea was not new.

For these reasons, Microsoft Corp. (“Microsoft” or “Petitioner”) petitions for *inter partes* review (“IPR”) of claims 1-50 (“Challenged Claims”) of U.S. Patent No. 9,456,053 (the “’053 patent”) (Ex-1001).

I. GROUNDS FOR STANDING

Petitioner certifies the '053 patent is available for IPR, and Petitioner is not barred or estopped from requesting IPR on the grounds herein. Petitioner files this Petition within one year of service of Patent Owner Sandpiper CDN, LLC's ("PO") district court complaint. Ex-1034.

II. IDENTIFICATION OF CHALLENGE

A. Prior Art

Petitioner's grounds rely upon the prior art below based on an assumed priority date of December 14, 2011. The references are analogous to the '053 patent and each other for being in the same field of endeavor (e.g., network servers and/or CDNs) or reasonably pertinent to the problems faced by the inventor (e.g., configuring CDN networks). *Lanning*, ¶¶69-70; *see also* ¶¶1-4, 17-32.

Prior Art¹
Lewin, U.S. Patent 7,010,578 filed September 21, 2000; issued March 7, 2006; prior art under at least §102(b) (Ex-1005)
Wein, U.S. Patent 7,240,100 filed April 16, 2001; issued July 3, 2007; prior art under at least §102(b) (Ex-1006)
Liu, U.S. Publication 2010/0257258 filed May 25, 2010; published October 7, 2010; prior art under at least §102(b) (Ex-1007)
Devanneaux, U.S. Publication 2007/0156845 filed December 29, 2006; published July 5, 2007; prior art under at least §102(b) (Ex-1008)
Nygren, <i>The Akamai Network: A Platform for High-Performance Internet Applications</i> , published July 2010 ² ; prior art under at least §102(b) (Ex-1009)
Lewin819, U.S. Publication 2010/0274819 filed June 29, 2010; published October 28, 2010; prior art under at least §102(b) (Ex-1010)
Crosbie, International Publication WO 2000/13456, filed August 31, 1999; published March 9, 2000; prior art under at least §102(b) (Ex-1011)

¹ Pre-AIA 35 USC §102. Lewin, Wein, Liu, Devanneaux, Lewin819 also qualify per AIA §§102(a)(1), (a)(2); Nygren, Crosbie per §102(a)(1).

² See Declaration of June Munford (Ex-1033) corroborating this date. Petitioner reserves the right to further corroborate and argue earlier dates in other forums.

B. Grounds for Challenge

Ground (§103)	Claims	Prior Art
1	1-4, 6, 8-9, 11-16, 21-23, 25, 27-30, 32, 34-35, 37-42, 47, 49-50	Lewin, Wein
2A	1-2, 4, 6, 9-16, 18-28, 30, 32, 35-42, 44-50	Liu, Devanneaux
2B	17, 43	Liu, Devanneaux, Nygren
2C	3, 5, 7-8, 29, 31, 33-34	Liu, Devanneaux, Lewin819
2D	18-20, 44-46	Liu, Devanneaux, Crosbie

III. TECHNOLOGY OVERVIEW

To provide end users access to content over the Internet—whether to load a website, or stream music or movies—content owners must typically maintain a server that stores their content library and transmits content from the library to the end users upon request.

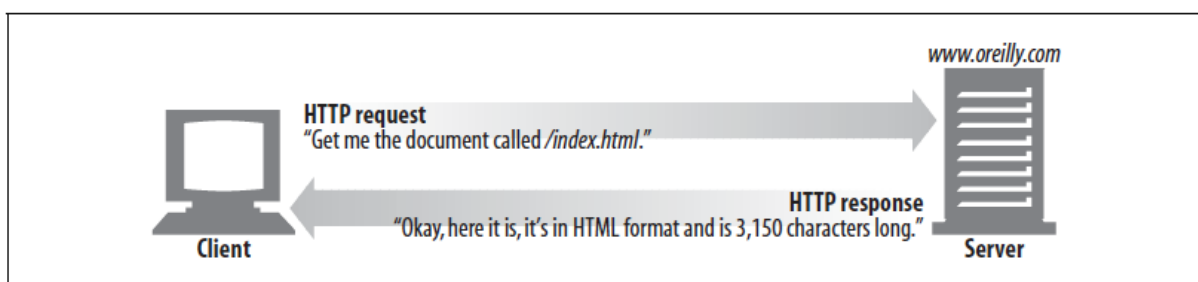


Figure 1-1. Web clients and servers

Gourley, 4; Lanning, ¶37.

However, as internet usage grew exponentially in the early 2000s, “serving Web content from a single location...present[ed] serious problems for site scalability, reliability, and performance,” as servers would overload and crash if they received too many requests at one time. *Dilley*, 50; *Lanning*, ¶46. Moreover, transmitting content across vast distances often took a significant amount of time. *Lanning*, ¶47. To solve these problems, companies began building CDNs that provided additional layers of geographically distributed servers between the end user’s device (or “client”) and the content owner’s origin server. *Id.* The CDN’s servers would “cache” (or store) copies of at least some content and respond to requests on behalf of the content owner, rather than send all requests to the origin server to fulfill. *Id.*

Because CDNs delivered content on behalf of content owners, both CDN operators and content owners needed a say in how content would be delivered; CDN operators needed to be able to add and remove servers from the network, and configure, e.g., how servers would interact with one another to distribute configurations and content across the servers, balance load across servers from filling end-user requests, as well as what type of data to collect about the network. *Dilley*, 52-54; *Nygren*, 6-7, 11, 15. Content owners, for their part, needed a say in how their own content would be handled by the network, as well as what type of data to collect about requests for their own content. *Dilley*, 52-54; *Nygren*, 5, 16. These network configurations could be controlled by a central control system, or

distributed throughout the network. *Dilley*, 54-56. And configuration information could be provided in standalone files or baked into communications between servers. *Nygren*, 14-15, *Gourley*, 142; *Lanning*, ¶53; *see also id.*, ¶¶33-56 (citing *Smith*, Abstract, Figs. 1, 4, ¶¶5-9; *Harvell*, Abstract, Fig. 1, ¶¶2-5; *Kovacs*, Abstract, 1:15-2:3, Fig. 1; *Lyon*, Abstract, Figs. 6A-C; *Outten*, Fig. 2).

IV. '053 PATENT OVERVIEW

A. Specification

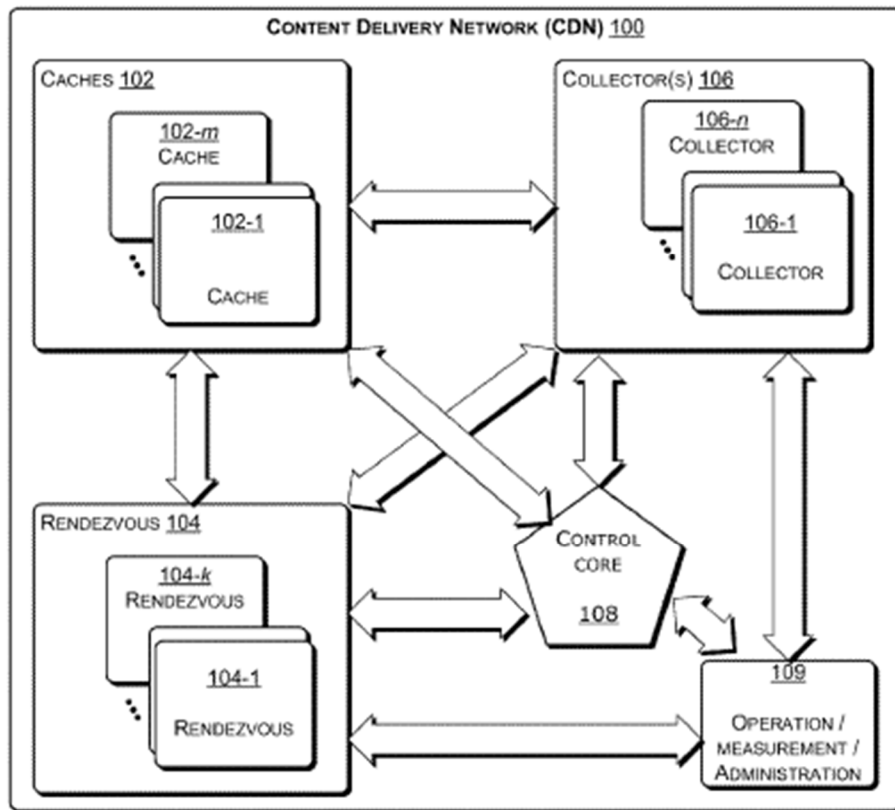
The '053 patent is directed to configuring CDNs, in which CDN operators control how the CDN is configured, generally, using “global” configurations, and customers control how the CDN handles their content, specifically, using “customer” configurations. *'053 patent*, 13:23-27, 13:32-36, 13:54-56, 14:14-32, 15:6-8, 37:26-39, 38:51-65. “Global” configurations may control things such as how the network operates and how it distributes load, and apply to *all* content, whereas “customer” configurations may control more discrete features when delivering a specific content owner’s content. *Id.* A “control core”—a centralized system—manages the distribution of these various configurations throughout the network. *Id.*, 4:43-50, 14:12-32, 18:26-31.

The '053 patent also discloses other elements, such as a “rendezvous mechanism” that controls how end-user requests are directed to particular servers to

fulfill requests for content, and “collectors” that collect logs on network operations.

Id., 3:24-32, 4:43-50, 40:47-41:21, 42:10-27.

Fig.1



Id., Fig. 1; *Lanning*, ¶¶57-60.

B. Prosecution History

During prosecution, the Examiner rejected all claims of the '053 patent in view of Lewin (U.S. Patent No. 7,010,578). Ex-1002, 523-40. In response, the Applicant amended the claims and argued that the Examiner's application of Lewin confused “customer” and “client” as specified in the claims, and the Examiner allowed the claims to issue thereafter without explanation. *Id.*, 592; *see also id.*, 592-601 (amended claims), 602-06 (PO arguments), 630-35 (allowance). But, as

demonstrated below, Lewin in fact taught “customer[s]” and “client[s]” consistent with the claims, and the Examiner also failed to combine Lewin with other readily available prior art, which would have supported a continued rejection. *Lanning*, ¶¶61-64.

V. LEVEL OF ORDINARY SKILL

A POSITA at the time of the invention 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 vice versa. *Lanning*, ¶¶65-67. Petitioner’s expert, Mark Lanning, was at least a POSITA as of the earliest possible priority date of the ’053 patent. *Id.*, ¶67; *see also id.*, ¶¶5-16.

VI. CLAIM CONSTRUCTION

No express constructions are necessary in this IPR; Petitioners have applied the claim terms’ ordinary meanings as understood by a POSITA.³ *Lanning*, ¶68.

³ Petitioners reserve the right to respond to PO/Board constructions. Petitioners reserve §112/claim scope arguments.

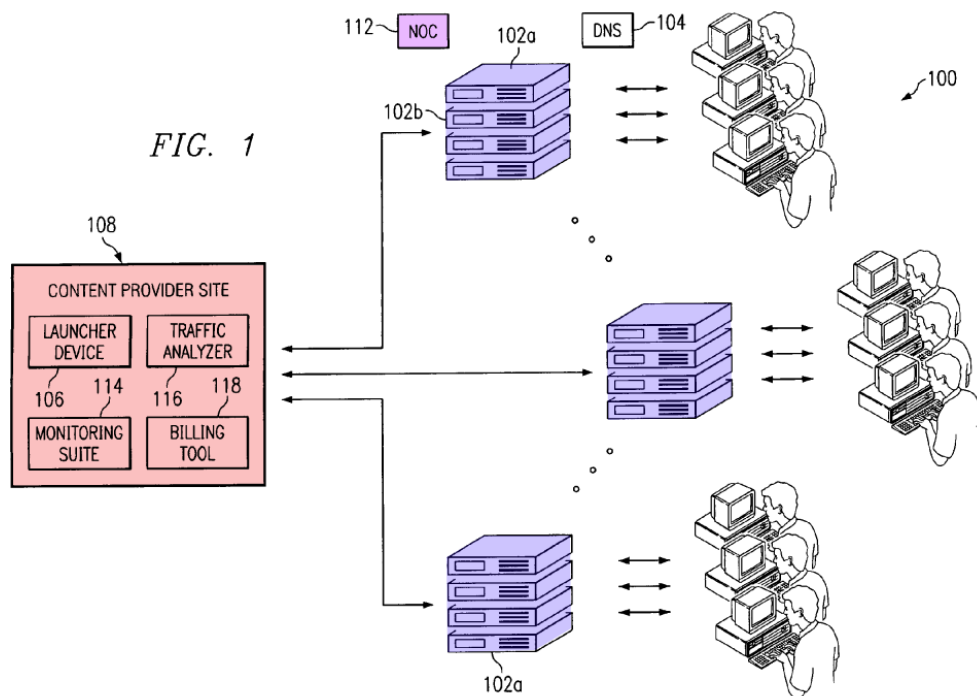
VII. DETAILED EXPLANATION OF GROUNDS

A. Ground 1: Obviousness over Lewin and Wein (Claims 1-4, 6, 8-9, 11-16, 21-23, 25, 27-30, 32, 34-35, 37-42, 47, 49-50)

1. Prior Art

a. Lewin

Lewin discloses “a content delivery service” (“CDS”) (*Lewin*, Abstract), “compris[ing] a preferably global content delivery network (CDN) 100 of content delivery servers 102a-n,” (*id.*, 2:47-50). Content providers (red) link their site to the CDN (blue, purple), which hosts and serves their website content on their behalf. *Id.*, 3:14-23, 6:9-17; *Lanning*, ¶71.



Lewin, Fig. 1.⁴

Lewin, however, specifically concerns allowing third parties *other than* the CDN operator to add their own caches (or servers) to participate in the CDN and “extend the reach of its network.” *Id.*, 1:61-2:12; *Lanning*, ¶72. When third parties add a cache to the CDN, the CDN sends the third-party cache configuration information defining how the CDN operates, and the third-party cache sends the CDN configuration information defining how “how [the third-party cache] will participate in the network (or not).” *Lewin*, 3:43-6:8; *Lanning*, ¶¶72-74.

Separately, Lewin explains how the network uses modified uniform-resource locators “URLs” (like www.google.com)—which Lewin calls “ARLs,” for “alternative resource locators”—to point clients “to the content delivery service network” rather than “the content provider’s site.” *Lewin*, 3:19-23; *Lanning*, ¶75.

b. Wein

Wein discloses the various configuration parameters involved in maintaining CDN networks, noting configurations are provided in “metadata.” *Wein*, Abstract, 1:17-52; *Lanning*, ¶76. Wein explains that the metadata provides control parameters that determine how the CDN content server handles an object request. *Wein*, 2:13-

⁴ Images annotated unless otherwise noted.

27. This metadata may either be provided within a request itself, or in a separate configuration file. *Id.*, 3:8-23; *Lanning*, ¶76.

2. Motivation to Combine Lewin with Wein

Although Lewin describes the high-level configurations involved in adding a third-party cache to an existing CDN network, Lewin does not expressly describe certain details as to how the primary CDN networks are configured by both CDN operators and customers to control the operation of the CDN and delivery of an owner's content. *Lanning*, ¶77.

A POSITA would have been motivated to combine the teachings of Lewin with Wein to understand how to configure the primary CDN, apart from configuring third-party caches added to the network. *Id.*, ¶78. For instance, Wein expressly describes how the primary CDN uses CDN-operator-provided configurations and customer-provided configurations, and details how ARLs—like those Lewin discloses—can also include customer-configuration information. *Wein*, 4:39-52, 4:62-5:15, 8:12-31; *Lanning*, ¶78. In other words, whereas Lewin is primarily concerned with configuring a third-party cache that is ***added*** to a CDN, Wein is primarily concerned with configuring the CDN itself. *Lanning*, ¶78.

A POSITA would have had a reasonable expectation of success in borrowing Wein's teachings for Lewin's network because they are both nearly identical systems. *Lewin*, 1:61-2:12, 3:24-58; *Wein*, 2:13-27; *Lanning*, ¶79. Lewin and Wein

are both also publications from Akamai Technologies (*Lewin*, Cover (assigned to Akamai Technologies); *Wein*, Cover (same)), which was one of the most well-known CDN providers. *Lanning*, ¶79. Moreover, a POSITA would have understood that their respective teachings complement each other, rather than expressly alter one another, because they are directed to configuring different aspects of the network. *Id.*, ¶79. In other words, modifying *Lewin* would have involved nothing more than coding the network to *also* operate consistent with *Wein* and its processes for configuring the primary CDN, while *Lewin* remains focused on configuring third-party caches. *Id.*

3. Claims 1, 27

a. 1(pre)/27(pre)

If limiting, *Lewin* discloses or suggests 1(pre)/27(pre). *Lanning*, ¶¶80-82.

Lewin discloses “a content delivery *method* operative in a *content delivery network*.”⁵ *Lewin*, cl. 1; *id.*, 2:47-50 (describing a content-delivery service that “comprises a preferably global content delivery network (CDN) 100 of content delivery servers 102a-n”).

A POSITA would have also understood that *Lewin* discloses that “content providers” (*customers*) may utilize a CDN to serve content (*resources*) on their behalf, consistent with conventional CDNs. *See, e.g., id.*, 3:14-23, 6:9-17

⁵ Italicized text is claim text unless otherwise noted.

(explaining content providers host a website that may point to content hosted by a CDN); *Lanning*, ¶82.

b. 1(a)/27(pre)(a)

Lewin discloses or suggests 1(a)/27(pre)(a). *Lanning*, ¶¶83-85.

Lewin discloses the CDN includes “network operations center (NOC) 112 for monitoring the network to ensure that key processes are running, systems have not exceeded capacity, and that sets of content servers (so-called regions) are interacting properly.” *Lewin*, 2:59-64, Fig. 1. Lewin also discloses a “registration server” that controls access to the CDN. *Id.*, 4:27-31.

A POSITA would have understood that Lewin’s NOC and registration server collectively constitute a *control core* as claimed because they generally control the CDN—whether its operation or access—consistent with the ’053 patent. *Id.*; *Lanning*, ¶85; ’053 patent, 4:43-44 (“The control core mechanism 108 controls operation of the CDN”).

c. 1(b)/27(pre)(b)

Lewin discloses or suggests 1(b)/27(pre)(b). *Lanning*, ¶¶86-88.

Lewin discloses that the CDN includes a set of third-party caches (*plurality of cache servers*) “to cache and deliver content.” *Lewin*, 3:24-26; *see also* §VII.A.3.a (1(pre)/27(pre)) (teaching *the CDN serving resources on behalf of customers*).

Lewin explains “the third party cache is a box purchased or otherwise obtained from a cache vendor...and deployed in the owner’s network.... When third party caches are joined into the content delivery service...content may be shared across the resulting composite network.” *Lewin*, 3:29-42. A POSITA would have therefore understood that Lewin’s third-party caches comprise computer hardware (*constructed/hardware*) and are programmed to serve content and interact with the CDN (*adapted/software*). *Lanning*, ¶88.

d. 1(b)(1)-1(b)(1)(1)/27(A)

Lewin, alone or combined with Wein, teaches or suggests 1(b)(1)-1(b)(1)(1)/27(A). *Lanning*, ¶¶89-94.

Lewin discloses that third-party caches are “first ‘registered’ with the content delivery network” (*upon joining the CDN*). *Lewin*, 3:43-4:7; *Lanning*, ¶90.

During registration, “[t]he CDN preferably provides cache 302 with given information,” including “a time-to-live (TTL) indication 318, which controls how long registration is valid,” and “configuration information including log recipient and frequency 320, RT (real-time) statistics recipient and frequency data 322, a map URL 324, and content parsing information” (collectively, *global-configuration data*). *Lewin*, 4:57-6:3; *Lanning*, ¶91.

Wein similarly teaches “[a] CDN-generated metadata configuration file [(*global-configuration file*)] may be distributed to all or a portion of the CDN servers, preferably by some central authority” (*control core*). *Wein*, 8:3-5; *Lanning*, ¶92.

A POSITA would have understood that both Lewin’s and Wein’s configuration information provided to the cache constitutes *global-configuration data* as claimed, because it establishes the cache’s presence and operation as part of the CDN, consistent with the ’053 patent. *Lanning*, ¶93; ’053 patent, 14:63-15:19. Although Lewin does not expressly describe which structure provides the configuration information to the third-party cache, a POSITA would have understood that either the NOC or registration server (*control core*) provides the configuration information, because these are the elements Lewin describes as controlling the operation of and access to the CDN. *Lanning*, ¶94. This is consistent with Wein, which expressly teaches using a “central authority” to push CDN-generated metadata configuration files. *Wein*, 8:3-5; *Lanning*, ¶94.

e. 1(b)(2)-1(b)(2)(1)/27(B)

Lewin, alone or combined with Wein, teaches or suggests 1(b)(2)-1(b)(2)(1)/27(B). *Lanning*, ¶¶95-98.

Lewin discloses that, during registration, some of the configuration data includes things like “[a] map URL” which “identifies a location of a network map within the CDN. Once the cache is configured into the CDN, it becomes a server in

the CDN network map and, as such, may assume responsibility...for delivering content” and has a time-to-live “TTL [that] is preferably long (e.g., one month).” *Lewin*, 5:7-12. Lewin also discloses the configuration information includes “content parsing data” which “is the information needed to identify CDN-provisioned content and to fetch it correctly.” *Id.*, 5:12-15. Lewin specifies one configuration parameter includes “long 1MapFrequency” which represents “[t]he number of seconds between Map updates.” *Id.*, 12 (Table).

Wein similarly teaches that “[a] metadata configuration file can be ‘pushed’ out to all servers *when it changes*,” and that the central authority “may change the CDN-generated metadata configuration file.” *Wein*, 8:3-8, 8:27-29⁶.

A POSITA would have understood based on Lewin, alone or combined with Wein, that after joining the CDN, third-party caches also receive updated global-configuration data (including, e.g., updated maps, parsing data, or other configurations) so that the third-party cache can account for any changes in how and where the CDN stores particular content to serve it to the client. *Lanning*, ¶97.

Finally, to the extent “selectively obtaining” (cl. 27) requires “pulling” updated global configuration data, a POSITA would have been motivated to configure Lewin’s system to pull such data. *Lanning*, ¶98. Wein, for instance,

⁶ All emphasis added unless otherwise noted.

expressly states that “[a] metadata configuration file can be ‘pushed’ out to all servers when it changes, or ‘pulled’ periodically from a set of intermediate staging servers,” (*Wein*, 8:5-10), as these were two of the most prominent ways to distribute information over a network. *Lanning*, ¶98 (citing *Reynolds*, ¶¶132-34; *Winter*, ¶103). Because Lewin is silent on the exact protocols used to distribute configuration information, a POSITA would have understood that either or both can be used, but would have appreciated that configuring the system to do both would provide added flexibility in how configuration information is distributed. *Lanning*, ¶98.

f. 1(b)(2)(2)/27(C)

Lewin, alone or combined with Wein, teaches several different configuration sets that constitute *customer-configuration information* as claimed in 1(b)(2)(2)/27(C). *Lanning*, ¶¶99-111.

First, Lewin discloses that, during registration, the “CDN preferably provides the cache given configuration information” including, e.g., “RT (real-time) statistics recipient and frequency data 322, a map URL 324, and content parsing information 326” after joining the network. *Lewin*, 4:57-67. When registered, “the cache may be provisioned to only get traffic from blocks of IP address that they ‘own,’” or “may be provisioned to not report the IP address of a requestor in a cache log.” *Id.*, 4:8-26. Although Lewin discloses an initial exchange of this information during registration, a POSITA would have appreciated that configuration files are routinely updated, and

thus the third-party cache may obtain updated configuration information from a customer after joining the CDN. *Id.*, 4:63-67 (configuration files includes, e.g., real time (RT) statistics), 5:4-8 (updating the RT statistics per time period), 5:35-36 (“the cache may need to re-register to keep itself current”); *Lanning*, ¶¶101-03.

Separately, Lewin explains how the system uses “ARLs”—or alternative resource locators—to point clients “to the content delivery service network” rather than “the content provider’s site.” *Lewin*, 3:19-23.

A POSITA would have understood that Lewin discloses two possible sources of *customer-configuration information*: (1) a configuration file (specifically for third-party caches); and (2) the ARL. *Lanning*, ¶104.

Configuration files. Lewin explains that the configuration file may be provided along with the third-party cache to specify how the cache interacts with the CDN. *Lewin*, 3:53-56, 4:8-26. Thus, when a CDN customer provides a third-party cache to supplement the CDN network—for instance, providing a cache at the customer’s corporate headquarters, or in a service region that has limited support for the customer’s clients by the CDN itself—the customer would provide the configuration file associated with that third-party cache. *Id.*; *Lanning*, ¶105.

A POSITA would have appreciated, however, that CDNs routinely incorporate additional customer-provided configuration files that allow customers to configure how a CDN will handle the customer’s content *within the CDN itself* (as

opposed to how the CDN will treat a third-party cache consistent with Lewin).

Lanning, ¶106. Wein, for instance, notes that CDNs may utilize “[a] per-customer metadata configuration file [that] may be located on the customer’s server to maximize a customer’s ability to customize the metadata configuration for the [content provider’s] CP’s objects.” *Wein*, 8:12-17; *see also id.*, 8:29-31 (“The content provider may customize the settings in the CP-generated configuration file”). “For example, there may be metadata settings that control logging of cookies or referrer field with customer logs. There may be metadata settings to require that CDN servers revalidate an object before it is served if the time-to-live has expired, or to serve a default object if the cached object cannot be revalidated.” *Id.*, 5:11-26.

A POSITA would have understood based on the teachings of Lewin and Wein that CDNs provide several avenues for customers to configure both the structure of the CDN and their content hosted by the CDN. *Lanning*, ¶107. Lewin alone teaches configuring third-party caches to interact with the CDN, but a POSITA would have understood that Lewin would have, or at a minimum been motivated to modify Lewin to also incorporate, customer-provided configuration files that control the handling of the customer’s content throughout the entire CDN, including third-party caches, as expressly taught by Wein. *Id.* A POSITA would have had a reasonable expectation of success in incorporating both types of configuration files because they

are directed to different aspects of the network (third-party caches versus customer content), and thus work alongside one another without conflict. *Id.*, ¶108.

ARLs. Although Lewin also mentions that the CDN utilizes ARLs to direct clients to the CDN, (*Lewin*, 3:19-23), a POSITA would have also understood that ARLs were a known term in the CDN arts that routinely have configuration information incorporated within them. *Lanning*, ¶109. Wein, for instance, explains how ARLs are used to point clients to the CDN (like Lewin), (*Wein*, 4:39-52), but also explains that “ARL[s] contain all of the required metadata ‘in-URL,’” in which the “metadata is the set of all control options and parameters that determine how a CDN content server will handle a request for an object” (*id.*, 4:62-5:15). Wein explains that configuration information in ARLs “is specific to the individual request the ARL issues,” whereas configuration information in configuration files “is more general than the other types,” and thus “generally the metadata contained in an ARL takes precedence over...metadata in configuration files.” *Id.*, 8:18-31. A POSITA would have therefore understood that Lewin’s ARLs would have, or at a minimum been motivated to modify Lewin’s ARLs to also incorporate, metadata (*customer-configuration information*) in Lewin’s ARLs to specify control parameters for handling the customer’s content in response to a request. *Lanning*, ¶111; §VII.A.2 (motivation).

g. 1(b)(2)(3)/27(D)

Lewin discloses or suggests 1(b)(2)(3)/27(D). *Lanning*, ¶¶112-14.

Lewin discloses a process of serving a customer's content to a client, which involves making a request of the customer's website, which then directs the client to the CDN, including third-party caches. *Lewin*, 3:7-42. Because Lewin, alone or combined with Wein, teaches that both the CDN and third-party caches have been configured (using *global-configuration data* and *customer-configuration information*) that specifies both URL maps and parsing information to locate the requested content, as well as configuration details as to how: (1) the third-party cache will handle certain requests; (2) how the CDN will handle certain requests; and (3) how specific content will be handled using ARLs, §§VII.A.3.d-VII.A.3.f (1(b)(1)-1(b)(2)(2)/27(A)-27(C)), a POSITA would have understood that the content would be served in accordance with each of the *global-configuration data* (e.g., configuring the CDN and engagement with third-party caches) and *customer-configuration information* (e.g., configuring how the CDN and/or third-party caches handle customer content). *Lanning*, ¶114.

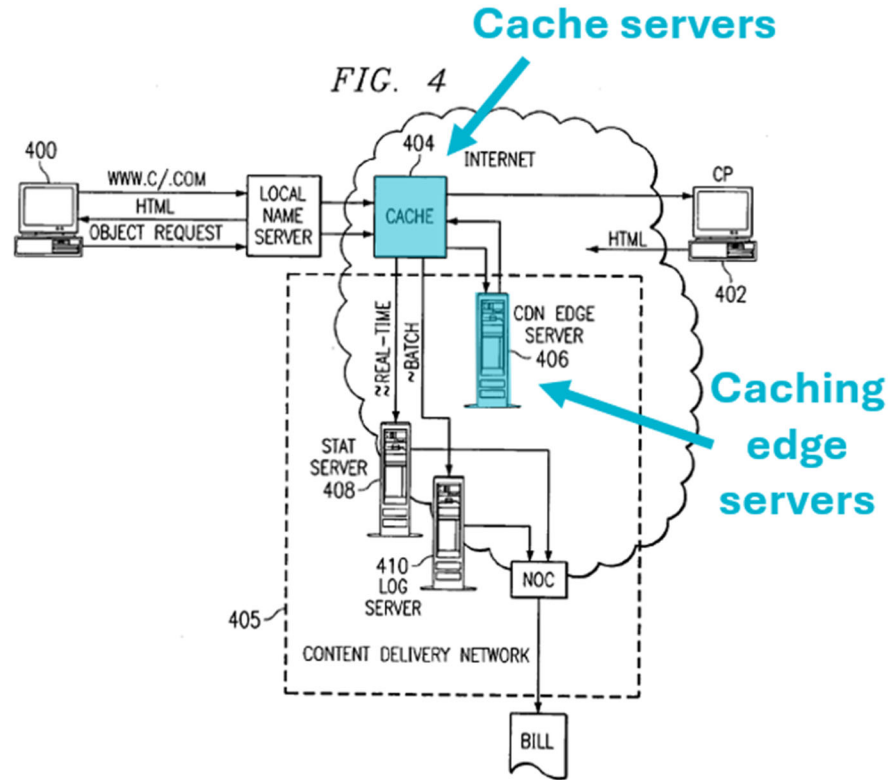
4. Claims 2/28

Lewin discloses or suggests claims 2/28's additional limitations. *Lanning*, ¶¶115-20.

Lewin discloses that, upon a third-party cache joining the CDN (cls. 2/28),⁷ the CDN is then used as “an authoritative surrogate content server, just like any other CDN content server.” *Lewin*, 3:49-53. A POSITA would have understood, however, that to act like “an authoritative surrogate content server,” the third-party cache would need to cache content to serve. *Lanning*, ¶¶116-17 (citing *Swildens*, ¶¶43, 130). A POSITA would have found it obvious, therefore, for Lewin’s third-party caches to obtain copies of content to cache upon joining the CDN, and would have obtained that content from other CDN servers already storing the content (*obtain data from one or more other locations in the CDN*). *Id.*, ¶¶117-18.

Indeed, Lewin discloses that, after a third-party cache joins the CDN (cl. 28), the cache becomes part of an “interconnection of caches” making up the CDN that “share[s] responsibility” for delivering content called, “content peering.” *Lewin*, Abstract. Lewin explains that when a third-party cache receives a request for content, “[i]f the requested object is not present in the third party cache 404, the cache preferably goes out to the CDN 405 (e.g., CDN edge server 406) to retrieve the object” (*obtain data from one or more other locations in the CDN*). *Id.*, 6:9-41.

⁷ Unlike Claims 1-2, Claims 27-28 do not distinguish between “upon joining” or “having joined” the CDN, and thus either circumstance is sufficient to meet its elements.



Id., Figs. 2, 4; *Lanning*, ¶119. A POSITA would have understood, therefore, that Lewin's third-party cache has hardware (e.g., *constructed*) and software (e.g., *adapted*) to physically and functionally connect to peer servers in the CDN to retrieve objects to serve to clients. *Lanning*, ¶120.

5. Claims 3/29

Lewin combined with Wein teaches or suggests claims 3/29. *Lanning*, ¶¶121-23.

Lewin combined with Wein teaches incorporating customer-configuration information associated with a specific customer in ARLs. §VII.A.3.f (1(b)(2)(2)/27(C)); *Lanning*, ¶122 Lewin also explains that content is served when

the cache receives an ARL. *Lewin*, 3:17-23, 6:42-7:28, Fig. 2. A POSITA would have therefore understood that because ARLs are used to request content from the CDN (or third-party cache), and customer-configuration information may be incorporated into the ARL, *Lewin* combined with *Wein* teach *obtaining customer-configuration information...upon receipt of a request for a resource associated with a specific customer*. *Lanning*, ¶123.

6. Claims 4/30

Lewin combined with *Wein* teaches or suggests claims 4/30. *Lanning*, ¶¶124-25.

Lewin combined with *Wein* teaches incorporating customer-configuration information associated with a specific customer in a configuration file. §VII.A.3.f (1(b)(2)(2)/27(C)); *Lanning*, ¶125. Although *Lewin* describes customer-configuration information related to third-party caches, a POSITA would have understood customer-configuration information may also be provided in a file to configure the CDN—and by extension, third-party caches added to the CDN—on how to serve content from a particular customer. §VII.A.3.f (1(b)(2)(2)/27(C)); *Lanning*, ¶125. As *Lewin* explains, these configuration files are exchanged upon adding a third-party cache to the CDN (and may be updated thereafter), (§§VII.A.3.e-VII.A.3.f (1(b)(2)-1(b)(2)(2)/27(B)-27(C))), and thus a POSITA would

have understood that the configuration files are processed *prior to receipt of a request for a resource associated with the specific customer*. *Lanning*, ¶125.

7. Claims 6/32

Lewin alone or combined with Wein teaches claims 6/32's additional limitations. *Lanning*, ¶¶126-28.

Lewin alone or combined with Wein teaches providing customer-configuration information in a third-party cache configuration file, a customer-provided configuration file, and/or an ARL. §VII.A.3.f(1(b)(2)(2)/27(C)); *Lanning*, ¶127. A POSITA would have understood that each of these constitute *CDN resources* as claimed because they are resources used by the CDN to deliver content. *Lanning*, ¶127; '053 *patent*, 15:66-16:2 (distinguishing between customer resources (e.g., content) and CDN resources).

Alternatively, because the customer-configuration information may specify how third-party caches engage with the CDN to fulfill requests, or specify how customer content is handled, a POSITA would have also understood the customer-configuration information includes information calling for how the third-party cache or CDN process and fulfill requests using CDN-processing resources. *Lanning*, ¶128; '053 *patent*, 40:39-40 (noting a customer-configuration script may “provide...for CDN resources.”).

8. Claims 8/34

Lewin alone or combined with Wein teaches or suggests claims 8/34's additional limitations. *Lanning*, ¶¶129-30.

Both Lewin and Wein teach that customer-configuration information may specify things such as traffic restrictions and registration and validation requirements. *See, e.g., Lewin*, 4:27-56; *Wein*, 4:62-5:10. Because operating based on these configuration details would require processing by the network (either the third-party cache or CDN) and relate to the cache's/CDN's ability to deliver customer content, a POSITA would have understood the customer-configuration information *specifies customer-specific processing requirements for resources served on behalf of that customer*. *Lanning*, ¶130; *see also* §VII.A.7 (cls. 6/32).

9. Claims 9/35

Lewin alone or combined with Wein teaches claims 9/35's additional limitations. *Lanning*, ¶¶131-33.

Lewin alone or combined with Wein teaches providing global-configuration information. §VII.A.3.d (1(b)(1)/27(A)); *Lanning*, ¶132. A POSITA would have understood that the global-configuration file constitutes a *CDN resource* as claimed because it is a resource used by the CDN to configure the CDN network. *Lanning*, ¶132; '053 *patent*, 15:66-16:2 (distinguishing between customer resources (e.g., content) and CDN resources).

Alternatively, because the global-configuration information specifies how the CDN network is arranged and how content is handled across the network, a POSITA would have also understood the global-configuration information includes information calling for how the third-party cache or CDN processes and fulfills requests using CDN-processing resources. *Lanning*, ¶133; '053 patent, 40:39-40 (noting a customer-configuration script may “provide...for CDN resources.”).

10. Claims 11/37

Lewin discloses or suggests claims 11/37's additional limitations. *Lanning*, ¶¶134-36.

Lewin discloses that, apart from the third-party cache, the CDN also includes other edge caches (*cache servers*). *Lewin*, 6:24-33 (noting third-party caches can fetch content from CDN “edge server[s]”), cl. 1 (a CDN “having a set of edge caches”), Figs. 1-4; *Lanning*, ¶135. A POSITA would have understood that, upon joining a CDN (cls. 2/28) and/or after joining the CDN (cl. 28), the third-party cache would obtain content from other caches in the CDN, either to host the content for itself in the first instance, or to fetch when it receives a request for content it is not storing. §VII.A.4 (cls. 2/28); *Lanning*, ¶136.

11. Claims 12/38

Lewin discloses or suggests claims 12/38's additional limitations. *Lanning*, ¶¶137-38.

Lewin explains that third-party caches “may ‘peer’ with the CDN and each other to allow participants to share their networks and content delivery resources.” *Lewin*, 8:63-67; *Lanning*, ¶138. A POSITA would have considered it obvious that, upon joining a CDN (cls. 2/28) and/or after joining the CDN (cls. 28), both third-party and edge cache servers would obtain content from peers in the CDN. *Lewin*, 8:63-67; §VII.B.4 (cls. 2/28); *Lanning*, ¶138.

12. Claims 13/39⁸

Lewin discloses or suggests claims 13/39’s additional limitations. *Lanning*, ¶¶139-41.

Lewin discloses the configuration information (*global-configuration data*) includes mapping and content-parsing data, which “identifies a location of a network map within the CDN” and “is the information needed to identify CDN-provisioned content and to fetch it correctly.” *Lewin*, 5:7-15; §§VII.A.3.d-VII.A.3.e (1(b)(1)-(b)(2)/27(A)-(B)). Lewin also discloses a server that receives a request contacts other servers when the receiving server does not have the requested content. *See, e.g., Lewin*, 6:42-7:28, Fig. 6. A POSITA would have understood that any server (e.g., a

⁸ Petitioner reserves the right to challenge the definiteness of “said information” in Claims 13/39 in other forums, but treats it as global configuration *data* for purposes of this petition consistent with Claims 1/27 and 2/28. *See, e.g., ’053 patent*, 7:51-55.

CDN or third-party cache) that receives this configuration information would *determine...locations using said information*, and particularly the location of content in another server in the network to fill requests. *Lanning*, ¶141.

13. Claims 14/40

Lewin alone or combined with Wein teaches claims 14/40's additional limitations. *Lanning*, ¶¶142-45.

A POSITA would have understood that caches receive configuration information either from the NOC or registration server (*control core*), (§VII.A.3.d (1(b)(1)/27(A)), which constitutes a location in the CDN. *Lanning*, ¶143.

Wein, however, also teaches that although using a central authority to distribute configuration information is preferred, *updated* configuration information can either be “pushed” (e.g., from the central authority) or “‘pulled’ periodically from a set of intermediate staging servers.” *Wein*, 8:5-8. A POSITA would have been motivated to implement any of these ways of communicating updated configuration information to servers within the CDN, as they were all well-known options to choose from. *Lanning*, ¶144. However, a POSITA would have appreciated that because of the importance of ensuring all servers have the most up-to-date configurations throughout what may be an expansive CDN, staging updated configuration information at intermediate servers (*a location in the CDN*) would

provide a fast and efficient way of transmitting the configuration information from the intermediate servers to the edge. *Id.*, ¶145.

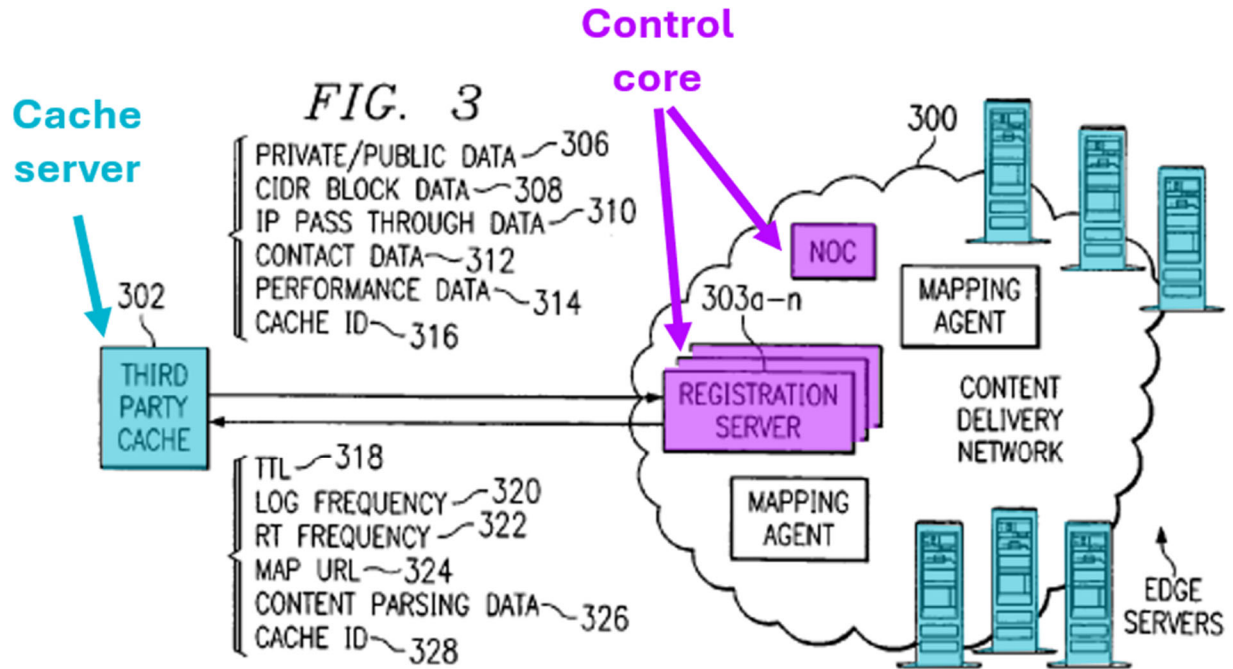
14. Claims 15/41

Lewin alone or combined with Wein teaches claims 15/41 by obtaining configuration information from the NOC/registration server (*control core*). §VII.A.13 (cls. 14/40); *Lanning*, ¶146.

15. Claims 16/42

Lewin discloses or suggests claims 16/42's additional limitations. *Lanning*, ¶¶147-48.

Lewin discloses a network operations center (NOC) and set of dedicated registration servers for registering third party cache servers in a CDN implemented as, e.g., Pentium-based machines running the Linux operating system and Apache web server software.



Lewin, Fig. 3; 2:44-64, 3:56-60; *see also* §VII.A.3.b (1(a)/27(pre)(a)); *Lanning*, ¶148.

16. Claim 21

Lewin discloses or suggests claim 21's additional limitations. *Lanning*, ¶¶149-51.

Lewin discloses that the CDN comprises "a dynamic domain name service (DNS) system," (*Lewin*, 2:47-51), and provides one example in which "the CDN may provision a given low level DNS server to manage content requests originating from a set of third party caches." *Lewin*, 7:57-60. A POSITA would have understood a DNS to manage content requests constitutes a *rendezvous mechanism* consistent with the '053 patent. *Lanning*, ¶¶150-51; '053 patent, 4:29-35 ("the rendezvous system 104 is used to direct client resource requests [and] is preferably implemented using the DNS").

17. Claims 22/47(pre)

Lewin discloses or suggests claims 22/47(pre)'s additional limitations. *Lanning*, ¶¶152-54.

The '053 patent uses collector mechanisms to collect information—e.g., performance data and logs—about content delivered in the CDN from the caches and report it to content providers via the control core. '053 *patent*, Figs. 1 and 21D, 41:10-18. Lewin discloses the CDS “may also include a logging mechanism to generate appropriate billing and reporting.” *Lewin*, Abstract. For instance, the third-party server sends the CDN NOC real-time statistics and log files including cache server health and usage statistics. *Id.*, 7:64-8:17. The cache uses a log server 410 to “dump” complete logs of content it has served on a periodic basis, or may send only expressly matched log entries in a secure and compressed format. *Id.*, Fig. 4, 8:18-32; *Lanning*, ¶153.

18. Claims 23/47(E)-(F)

Lewin discloses or suggests claims 23/47(E)-(F)'s additional limitations. *Lanning*, ¶155.

Lewin's discloses that the third-party cache servers are *constructed and adapted to* generate reports on cache server health and usage statistics (*generate information*) which is then sent to the logging mechanism (*provide...some of said*

generated information to said collector system). §VII.A.17 (cls. 22/47(pre)); *Lanning*, ¶155.

19. Claim 49

Lewin combined with Wein teaches claim 49. *Lanning*, ¶¶156-57.

When implementing a CDN that utilizes customer-specific configuration files, a POSITA would have understood that the CDN, or third-party caches connected to the CDN, would receive configuration information from multiple customers whose content is served by each respective cache, and that content would be served according to the global-configuration data (applied to all customers) and the customer-specific configuration provided by the customer whose content is being served. §§VII.A.3.d-VII.A.3.g (1(b)(1)-1(b)(2)(2)/27(A)-27(D)); *Lanning*, ¶157.

20. Claims 25/50

Lewin alone or combined with Wein teaches claims 25/50's additional limitations. *Lanning*, ¶158.

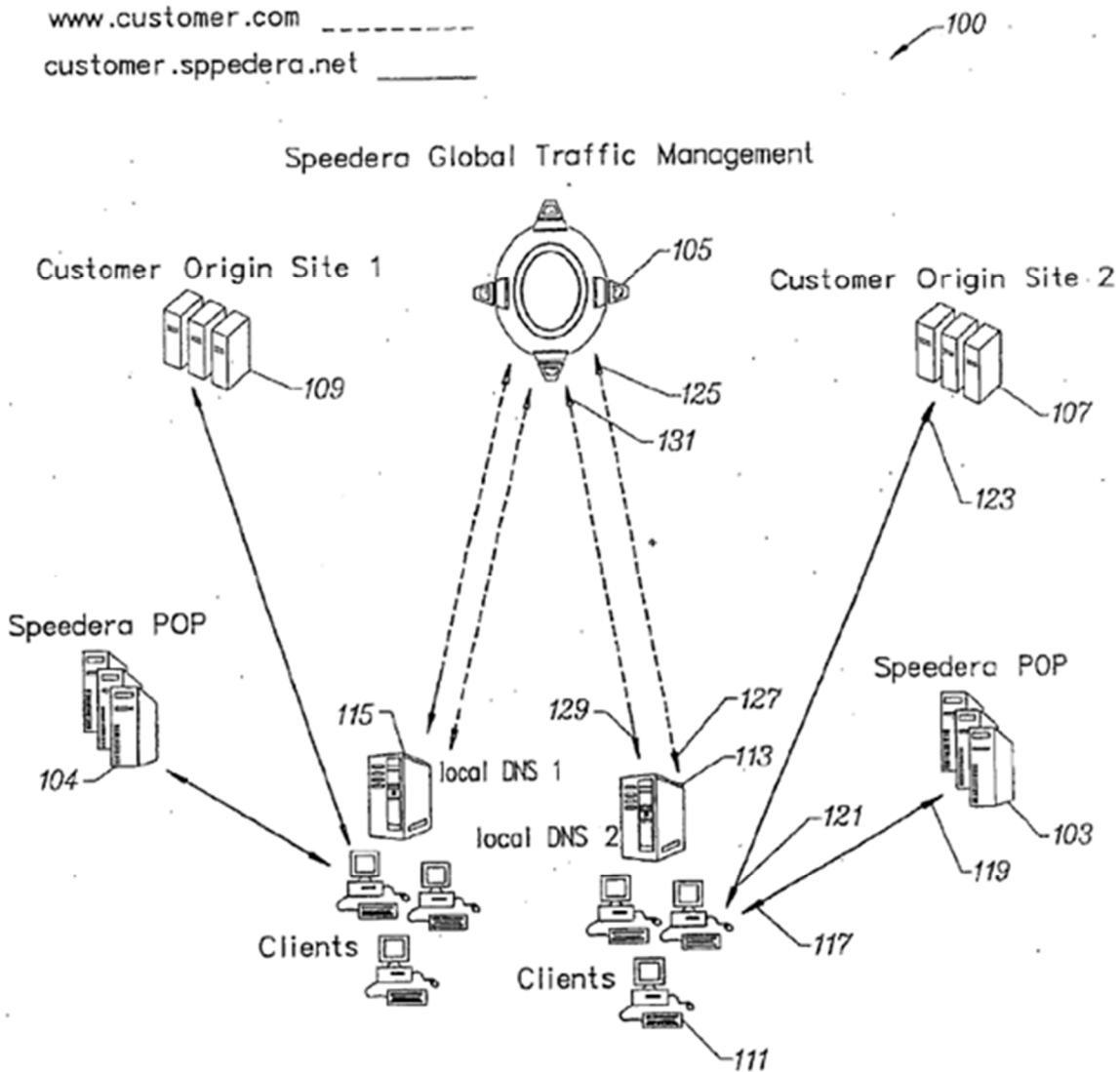
As described above, Lewin's caches (with or without Wein) serve content according to *global* and *customer configuration information* that is associated with the CDN and stored in copies of *configuration* files on each cache in the CDN. *Id.*; §§VII.A.3.f-VII.A.3.g (1(b)(2)(2)-(3)/27(C)-27(D)).

B. Ground 2A: Obviousness over Liu and Devanneaux (Claims 1-2, 4, 6, 9-16, 18-28, 30, 32, 35-42, 44-50)

1. Prior Art

a. Liu

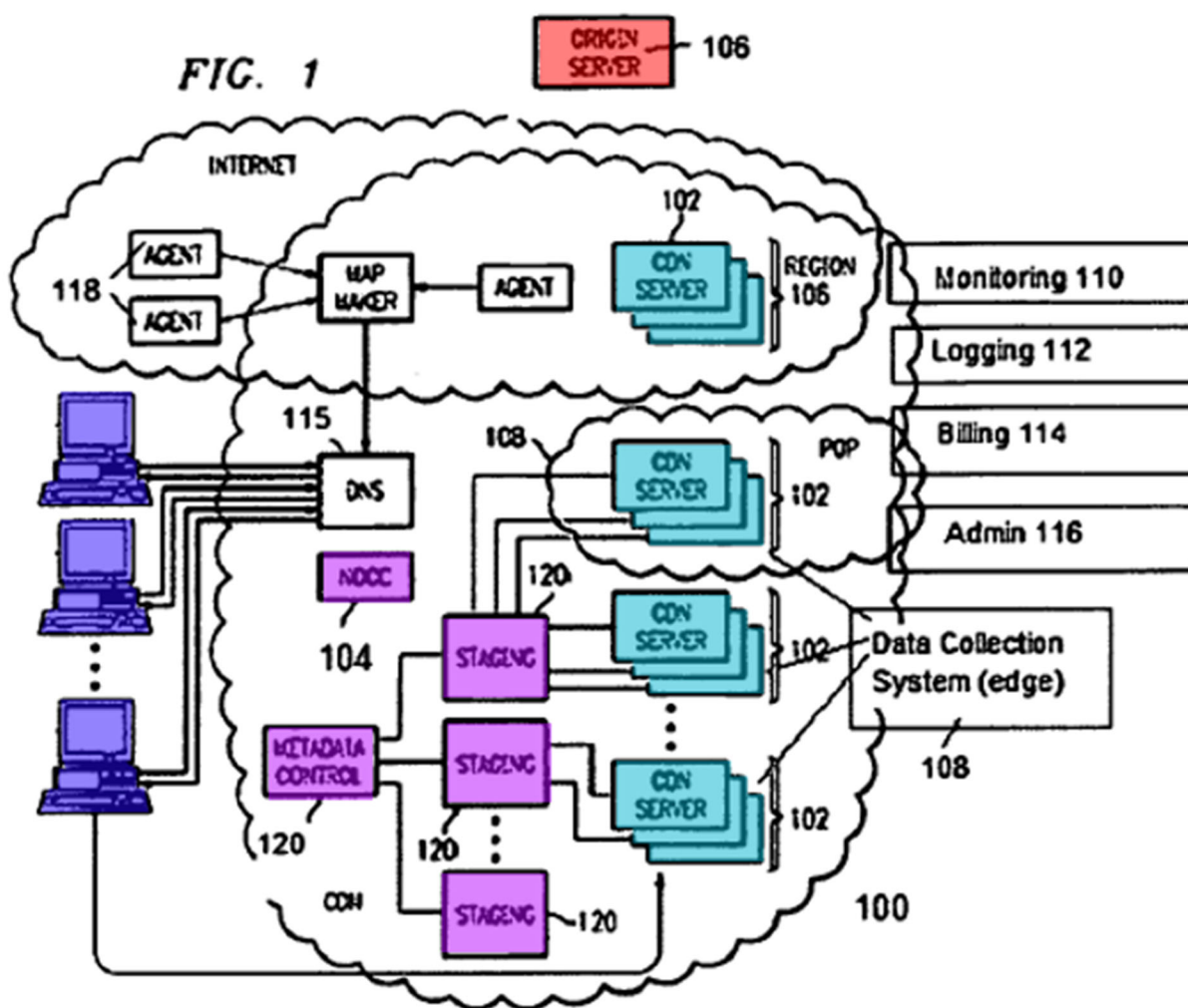
Liu discloses a CDN—specifically, a well-known one called “Speedera”—with a distributed, scalable domain-name system that helps coordinate delivery of requests to the CDN and prevents network attacks. *Liu*, Abstract, ¶¶22-23; *Lanning*, ¶159. Liu describes the network as a “Universal Delivery Network” (“UDN”) that couples a CDN with a global-traffic-management network (“GTM”) to manage the system as a whole:



Liu, Fig. 1, ¶28; *Lanning*, ¶159. *Liu*'s UDN includes origin servers and cache servers to store content. *Liu*, ¶¶29-30, 72; *Lanning*, ¶159. *Liu* also discloses providing "main" and "client" configuration files to configure the network and handling of content to manage network load-balancing and failure avoidance. *Liu*, ¶¶184, 192, 205-06; *Lanning*, ¶159.

b. Devanneaux

Devanneaux discloses another CDN—specifically, a well-known “Akamai” CDN created by the company Akamai Technologies—that includes staging servers 102, origin server 106, edge servers 106, and Network Operations Command Center (“NOCC”) 104 that manages the CDN using metadata controls, as well as other monitoring, logging, billing, administrative, data collection systems to route content to end users (blue):



Devanneaux, Fig. 1, ¶¶8-9; *Lanning*, ¶160. Specifically, Devanneaux configures its edge servers with metadata tags, configuration files, and scripts on a “customer-specific basis.” *Devanneaux*, ¶11; *Lanning*, ¶160.

2. Motivation to Combine Liu with Devanneaux

Liu discloses a UDN network that uses different sets of configuration files to manage the network. *See, e.g., Liu*, ¶¶205-06, 295-96. For instance, Liu explains that every network element in the system receives a copy of a “global configuration” or “Speedera configuration” file. *Id.*, ¶¶295-96, 301 (“At any time, a new configuration file can be pushed to all machines that need it”). Moreover, content providers (i.e., CDN customers) are provided tools to control their own configurations. *Id.*, ¶¶82, 90-96, 269, 428, Fig. 5A (noting customers can update content and invalidate old content, or control “DNS configuration parameters”). Liu, however, does not provide extensive details on the various types of configurations—particularly configurations available to content providers—used to control many aspects of the CDN networks. *Lanning*, ¶161.

Other prior art, such as Devanneaux, provided additional configurations for CDN networks generally, and cache servers specifically—including configuration files provided by the CDN network operator *and* customer—to manage the network while giving customers some control over how their content is handled by a CDN. *See, e.g., Devanneaux*, Abstract, ¶10 (“A cache hierarchy...is then controlled at a

fine-grain level using edge server and parent server configuration rules”), ¶21 (“the CDN edge servers are configurable to provide...delivery features on a customer-specific, customer domain-specific” level “to determine how it will handle...request[s],” and other configurations can be provided to “control how a given edge server establishes and maintains connections with one or more other edge servers.”).

A POSITA would have been motivated to implement additional types of configurations in Liu’s system, like those taught by Devanneaux, to provide more-granular control over both the network and how customer-specific content is handled by the network. *Lanning*, ¶¶162-63. For instance, although Liu focuses on controlling the configuration of the DNS servers that directs requests to cache servers, a POSITA would have recognized that the cache servers also need to be configured to establish their network relationships with one another and specify how content will be stored and delivered across the network. *Id.*, ¶163. A POSITA would have appreciated that incorporating additional configurations into Liu would provide both CDN operators with more flexibility to control their network—e.g., by specifying things such as “how a given edge server establishes and maintains connections with one or more other edge servers,” (*Devanneaux*, ¶21)—while also providing customers features such as prefetching content from different servers, or

other controls to specify how their content should be handled by the network (*id.*, ¶22; *Lanning*, ¶163).

A POSITA would have had a reasonable expectation of success in combining Devanneaux's added configurations with Liu's UDN/CDN network because a POSITA would have appreciated that Liu's network would have already had at least some cache-server configurations to control the arrangement of the UDN/CDN cache servers, even though Liu focused on the DNS server. *Lanning*, ¶164. All Devanneaux provides is an express disclosure of one easily implementable way to provide those cache-side configurations using XML language and scripts, (*Devanneaux*, ¶¶21-23, 38)—a widely known and commonly used computer-implemented language—that a POSITA would have understood could be easily implemented in Liu. *Lanning*, ¶164. Moreover, a POSITA would have known that Speedera and Akamai networks had similar structures,⁹ and thus improvements to one were generally transferable to the other. *Id.* In other words, the modification would have simply required ensuring Liu's cache servers were capable of parsing XML language as configuration instructions, and then providing configuration files

⁹ In fact, Akamai merged with Speedera in 2005, around the same time as Liu and Devanneaux. *See generally Mears*.

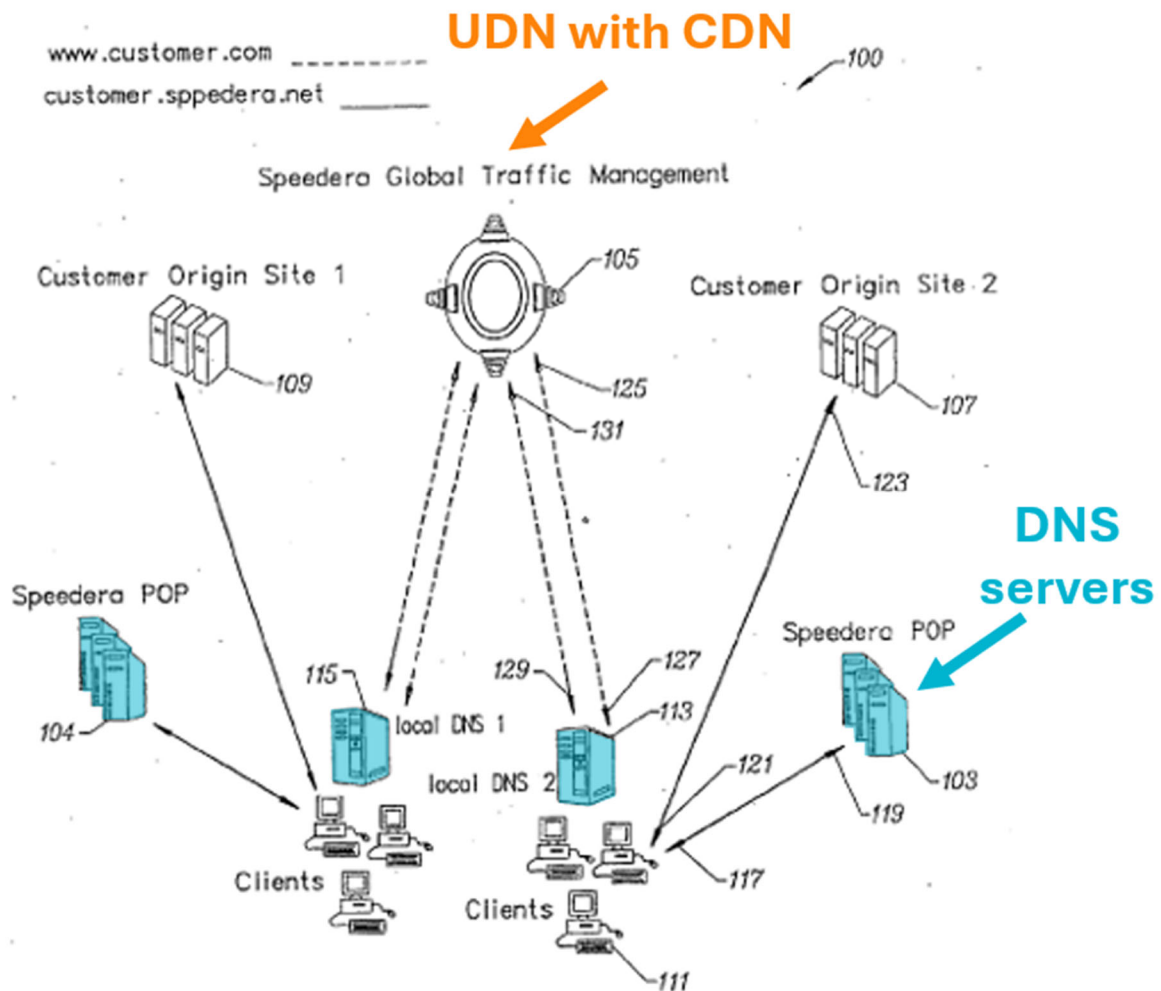
using XML. *Id.* Such an implementation would have been well within the skills of a POSITA. *Id.*

3. Claims 1, 27

a. 1(pre)/27(pre)

If limiting, Liu discloses or suggests 1(pre), 27(pre). *Lanning*, ¶¶165-67.

Liu discloses a “*computer-implemented method*” as part of a universal-delivery network (*content-delivery network*) that includes a CDN-cache sub-network and global-traffic-management sub-network. *Liu*, cl. 1, ¶¶22, 69, 260. Content providers pay for Liu’s UDN (*customers of the CDN*) to cache and serve their content, including websites, images, video, sound, and software (*serving resources*). *Id.*, ¶¶22, 25, 27-29; *Lanning*, ¶166.



Id., Fig. 1, ¶¶5, 8, 28, 31-39; *Lanning*, ¶166.

b. 1(a)/27(pre)(a)

Liu alone or combined with Devanneaux teaches or suggests 1(a)/27(pre)(a).
Lanning, ¶¶168-71.

Liu discloses that “[t]he configuration of the Speedera Network is maintained by centrally managed configuration files.” *Liu*, ¶295. Although Liu does not specify the exact system that “centrally manage[s]” the configuration files, a POSITA would have understood one or more systems operate together to do so, which would

constitute a *control core* as claimed. *Lanning*, ¶169; '053 patent, 4:43-44. For instance, Liu discloses that “[t]he network includes a Network Operations Center (NOC),” which contains many different centrally managed tools, like “NetView, AdminTools and LogServer software.” *Liu*, ¶¶70, 262. A POSITA would have understood, for instance, that the NOC would be a natural and obvious place to house the management for the configuration files. *Lanning*, ¶169.

This is consistent with other prior art, such as Devanneaux. Devanneaux discloses another CDN that uses a “distributed data transport mechanism...to distribute control information (e.g., metadata to manage content, to facilitate load balancing, and the like) to the edge servers.” *Devanneaux*, ¶9, Fig. 1.

A POSITA would have understood, therefore, that Liu’s UDN would have a similar computer-based system (such as a NOC-based system) to control and distribute the “centrally managed” configuration files. *Lanning*, ¶¶170-71.

c. 1(b)/27(pre)(b)

Liu discloses or suggests 1(b)/27(pre)(b). *Lanning*, ¶¶172-74.

Liu discloses that in a CDN, “web cache servers are used to serve content to clients,” (*Liu*, ¶22), and Liu’s UDN includes both cache servers for content as well as “DNS cache servers” for DNS content (*a plurality of cache servers*) (*see, e.g., id.*, ¶¶23, 30, 51-54, 71, 79-82, 305, 414-16, 420, 423-24, Figs. 1, 8; *Lanning*, ¶173). *See also* §VII.B.3.a (1(pre)/27(pre)) (teaching *the CDN serving resources on behalf of*

customers). Liu discloses “[t]he Speedera Network may consist of a number of Linux machines running Speedera software” (*Liu*, ¶240), and provides some descriptions of how caches have hardware (e.g., “memory” and “disk”) as well as software (e.g., “WebCache”). *Liu*, ¶¶305, 17-18; *see also id.*, ¶39 (“The steps can be performed using computer software or hardware or a combination”). A POSITA would have therefore understood that Liu’s caches (both web caches and DNS caches) comprise computer hardware (*constructed/hardware*) and programmed to serve content and interact with the CDN (*adapted/software*). *Lanning*, ¶174.

d. 1(b)(1)-1(b)(1)(1)/27(A)

Liu discloses or suggests 1(b)(1)-1(b)(1)(1)/27(A). *Lanning*, ¶¶175-77.

Liu discloses that “[t]he configuration of the Speedera Network is maintained by centrally managed configuration files” also known as “global configuration files or ‘Speedera configuration’ files [(*global-configuration data*)]. Every server in the network that needs configuration information has a copy of the appropriate current Speedera configuration file.” *Liu*, ¶¶295-300; *see also id.*, ¶¶205-06. Liu also explains that “[a]t any time, a new configuration file can be pushed to all machines that need it.” *Id.*, ¶301; *see also id.*, ¶¶40, 184 (noting DNS servers also use the configuration files to understand “how to weight the various criteria available when making decisions”); *Lanning*, ¶176.

A POSITA would have understood that, when a server first joins a network, it would obtain a copy of the global-configuration file (*upon joining the CDN...obtain global-configuration data*) to obtain, e.g., the network map and servers present at each network site (or “point-of-presence” (“POP”)), the list of hostnames serviced by the Speedera Network, and maps between those hostnames and the content to be served. *Liu*, ¶¶205, 295-301, 335; *Lanning*, ¶177. Otherwise, the new server would not know how to interact with the other servers in the network. *Lanning*, ¶177. Moreover, a POSITA would have also understood that the server would obtain that configuration file from the location in which the configuration file is being “centrally managed,” such as the NOC (*control core*) to ensure the most-recent version of the configuration file is obtained. §VII.A.3.b (1(a)/27(pre)(a)); *Lanning*, ¶177.

e. 1(b)(2)-1(b)(2)(1)/27(B)

Liu discloses or suggests 1(b)(2)–1(b)(2)(1)/27(B). *Lanning*, ¶¶178-80.

Liu discloses that “[a]t any time, a new configuration file can be pushed to all machines that need it.” *Liu*, ¶301; *see also id.*, ¶40 (noting DNS servers also use the configuration files to understand “how to weight the various criteria available when making decisions”); *Lanning*, ¶179. A POSITA would have therefore understood that Liu’s global-configuration file may be updated, and the updated configuration

file transmitted to all servers that need it (*having joined the CDN...[selectively] obtain updated global-configuration data [if needed]*). *Id.*

To the extent “selectively obtaining” (cl. 27) requires “pulling” updated global configuration data, a POSITA would have been motivated to configure Liu’s system to pull such data. *Id.*, ¶180. Pushing and pulling were two of the most prominent ways to distribute information over a network. *Id.* (citing *Reynolds*, ¶¶132-34; *Winter*, ¶103; *Wein*, 8:5-10; *Devanneaux*, ¶37). Configuring Liu’s servers to pull global configuration data rather than just push it would have been a simple substitution of known elements (pulling protocols versus pushing protocols), or applying one known technique (pulling) as a suitable alternative for another (pushing) to achieve the same, predictable result (updating configuration files). *Intel Corp. v. PACT XPP Schweiz AG*, 61 F.4th 1373, 1380 (Fed. Cir. 2023); *Snap Inc. v. You Map, Inc.*, No. 2024-112, 2025 WL 2793538, *3-4 (Fed. Cir. Oct. 1, 2025); *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 415-421 (2007). But a POSITA would have understood that either or both can be used, and would have appreciated that configuring the system to do both would provide added flexibility in how configuration information is distributed. *Lanning*, ¶180.

f. 1(b)(2)(2)/27(C)

Liu alone or combined with Devanneaux teaches or suggest 1(b)(2)(2), 27(C). *Lanning*, ¶¶181-84.

Liu discloses that customers may configure several parameters to control the delivery of their content. *See, e.g., Liu*, ¶¶82, 90-96, 269, 428, Fig. 5A (noting customers can update content and invalidate old content, or control “DNS configuration parameters”). A POSITA would have understood that these controllable parameters constitute *customer-configuration information associated with [at least one/a particular] customer of the CDN*. *Lanning*, ¶182.

Although Liu discloses customer-configurable parameters, a POSITA would have also been motivated to expressly implement in Liu’s network, and had a reasonable expectation of success in implementing, customer-configuration files such as those taught by Devanneaux. §VII.B.2 (motivation). Specifically, Devanneaux discloses using “configuration files” that provide customers added features “on a domain-specific, customer-specific basis” to provide greater control over how their content is handled on the network, such as content pre-fetching parameters and optimizations for traffic paths and client-server connections. *Devanneaux*, Abstract, ¶¶11, 21-23; *Lanning*, ¶183.

Upon incorporating customer-configuration files (*customer-configuration information*), a POSITA would have appreciated that Liu’s servers would be configured to obtain the configuration files associated with particular customers to determine the customer-defined rules for handling their content, just as Liu does with the global-configuration files. *Lanning*, ¶184.

g. 1(b)(2)(3)/27(D)

Liu alone or combined with Devanneaux teaches or suggests 1(b)(2)(3)/27(D).
Lanning, ¶¶185-89.

Liu discloses that DNS servers use the global-configuration file to determine “how to weight the various criteria available when making [a] decision” for how to direct a request for content to a web cache, (*Liu*, ¶40), and the global-configuration file also instructs network elements about the other servers in the network and “maps [customer’s] hostnames to the servers that can serve the content for that hostname,” among other settings, (*id.*, ¶205). Moreover, customers may configure valid content/URLs within the CDN. *Id.*, ¶¶90-96, 269. In other words, Liu’s global-configuration file controls how requests are handled and how content is served from the UDN, and the customer configurations control the active set of content that can be served. *Lanning*, ¶186. A POSITA would have understood that when a client requests content of a particular customer, the UDN would *serve to a client* the requested resource (*a particular resource associated with a particular customer*) *in accordance with (i) the global-configuration data, and (ii) customer-configuration information*. *Lanning*, ¶187.

Similarly, Devanneaux discloses the customer-controllable configuration files provide “[c]ontent handling rules” and “content control metadata” such as content pre-fetching parameters and optimizations for traffic paths and client-server

connections. *Devanneaux*, Abstract, ¶¶11, 21-23. In other words, Devanneaux's customer-configuration file controls how requests for each customer's content are handled and served from the CDN. *Lanning*, ¶188.

Upon combining Liu and Devanneaux, a POSITA would have similarly appreciated that content would be served to clients *in accordance with (i) Lui's global-configuration data, and (ii) Devanneaux's customer-configuration information*, because each provides separate rules to control the interactions over the network to serve each customer's content to clients. *Lanning*, ¶189.

4. Claims 2/28

Liu discloses or suggests claims 2/28's additional limitations. *Lanning*, ¶¶190-95.

Liu discloses that each server of a plurality of servers (e.g., web or DNS cache servers) has one or more UDN-specific software packages installed on it to operate as part of the UDN. *Liu*, ¶¶258-59. Liu also explains that the UDN uses software "to support publishing files and synchronizing them to many locations." *Id.*, ¶¶247, 79. And Liu also explains how DNS servers have sets of records "that are replicated on multiple servers" and "transfer zones from each other to maintain the consistency of information in the DNS." *Id.*, ¶6.

A POSITA would have understood that, when adding a web cache or DNS cache server to the UDN (*upon joining the CDN*, cls. 2/28)¹⁰, the new server would need not only the relevant configuration files, but also copies of the required software and web or DNS content it is supposed to host. *Lanning*, ¶¶192-94 (citing *Swildens*, ¶¶43, 130). A POSITA would have understood that either this additional software and content data can be pre-installed on the new server, or can be downloaded through the network from other parts of the UDN (e.g., the central NOC and/or other web/DNS cache servers storing web/DNS content), either of which would have been obvious (*constructed and adapted to...obtain data from one or more other locations in the CDN*). *Id.*

Liu also discloses that, after joining the UDN (cl. 28), if a server receives a request for web content that it does not have it can “request the content from the origin server” (*obtaining data from one or more other locations in the CDN*). *Liu*, ¶424; §VII.B.4 (cl. 28); *Lanning*, ¶195.

5. Claims 4/30

Liu alone or combined with Devanneaux teaches or suggests claims 4/30’s additional limitations. *Lanning*, ¶¶196-98.

¹⁰ See *supra* note 7.

Both Liu and Devanneaux teach that a cache server obtains customer-configuration information that is predetermined by the customer, and it would have been obvious to implement Devanneaux's specific customer-configuration information in Liu's system. §VII.B.3.f (cls. 1(b)(2)(2)/27(C)). A POSITA would have understood that each relevant cache obtains a copy of this configuration information independent of any requests for resources associated with the customer—either upon joining the network or as updates thereafter—and thus these configurations may be received both *prior to receipt of **any** request for a resource* (for configurations received upon joining the network), or for any requests received after the configurations are updated (*prior to receipt of a request*). *Lanning*, ¶¶197-98.

6. Claims 6/32

Liu alone or combined with Devanneaux teaches or suggests claims 6/32's additional limitations. *Lanning*, ¶¶199-201.

Liu alone or combined with Devanneaux teaches providing global-configuration files and customer configurations (Liu), as well as customer-configuration files (Devanneaux). §VII.B.3.f (cls. 1(b)(2)(2)/27(C)); *Lanning*, ¶200. A POSITA would have understood that each of these constitute *CDN resources* as claimed because they are resources used by the CDN to deliver content. *'053 patent*,

15:66-16:2 (distinguishing between customer resources (e.g., content) and CDN resources); *Lanning*, ¶200.

Alternatively, because the customer-configuration information may specify how the web and DNS servers work together to fulfill requests, or specify how customer content is handled (*Liu*, ¶¶184, 205, 295-301, 415; *Devanneaux*, ¶¶11, 20-22), a POSITA would have also understood the customer-configuration information includes information dictating how the CDN servers utilize CDN-processing resources to direct requests and deliver content. '053 *patent*, 40:39-40 (noting a customer-configuration script may “provide...for CDN resources.”); *Lanning*, ¶201.

7. Claims 9/35

Liu discloses or suggests claims 9/35's additional limitations. *Lanning*, ¶¶202-04.

Liu discloses providing global-configuration information. §VII.B.3.f (cls. 1(b)(2)(2)/27(C)); *Lanning*, ¶203. A POSITA would have understood that the global-configuration file itself constitutes a *CDN resource* as claimed because it includes “all the static information about the Speedera Network” and parameters used to control the CDN's operations, e.g., fine-tune the load balancing between cache servers. *Liu*, ¶205; *Lanning*, ¶203; '053 *patent*, 15:66-16:2 (distinguishing between customer resources (e.g., content) and CDN resources).

Alternatively, because the global-configuration information specifies how the CDN network is arranged and how content is handled across the network, a POSITA would have also understood the global-configuration information includes information calling for how the CDN servers process and fulfill requests using CDN-processing resources. *Liu*, ¶¶206, 211-13; *Lanning*, ¶204; '053 patent, 15:66-16:2 (distinguishing between customer resources (e.g., content) and CDN resources); 40:39-40 (noting a customer-configuration script may “provide...for CDN resources”).

8. Claims 10/36

Liu discloses or suggests claims 10/36's additional limitations. *Lanning*, ¶¶205-08.

Liu discloses that global configuration data is provided in a “global configuration file,” which “contains all the configuration information for that portion of the network, including: [(1)] List of servers allowed to change the configuration[; (2)] List of domains the network is responsible for[; (3)] List of services the machine in each POP supports[; and (4)] List of probes that perform latency checks at each POP.” *Liu*, ¶¶295-301; *see also id.*, ¶¶257, 314.

A POSITA would have understood *Liu*'s “global configuration file” constitutes and comprises a “global configuration object” consistent with the '053 patent, because it is a single, modular file with the required configuration data

contained within it, based on which the network acts. *Lanning*, ¶207; '053 patent, 14:24-28 (“The new cache may, e.g., request a single configuration **object** from the control core, and that configuration **object** may, itself, include the URLs for other configuration objects needed by the cache.”), 28 (Table; defining “Partial Object” as “Assembling separately-fetched parts of the same object into a complete object”), 37:40-41 (noting in one embodiment, the global-configuration object “acts as a ‘white list’ carrying valid protocols, hostnames and path prefixes”).

To the extent the global-configuration object is a discrete piece of the configuration file—like a table—used to inform content request routing, Liu teaches this as well, because Liu provides discrete pieces of information—such as the lists of machines and supported services—used for content request routing. *Liu*, ¶¶295-301; *see also id.*, ¶¶257, 314; '053 patent, Fig. 19 (1912) (a discrete component of the global-configuration data), 24:20-21 (“a table (the GCO) is used...in conjunction with the apparent target of the request”); 15:64-16:4, 37:26-39, 40:35-38 (a GCO is a configuration object that must be valid and determines whether a requested resource can be served); *Lanning*, ¶208.

9. Claims 11/37

Liu discloses or suggests claims 11/37’s additional limitations. *Lanning*, ¶¶209-11.

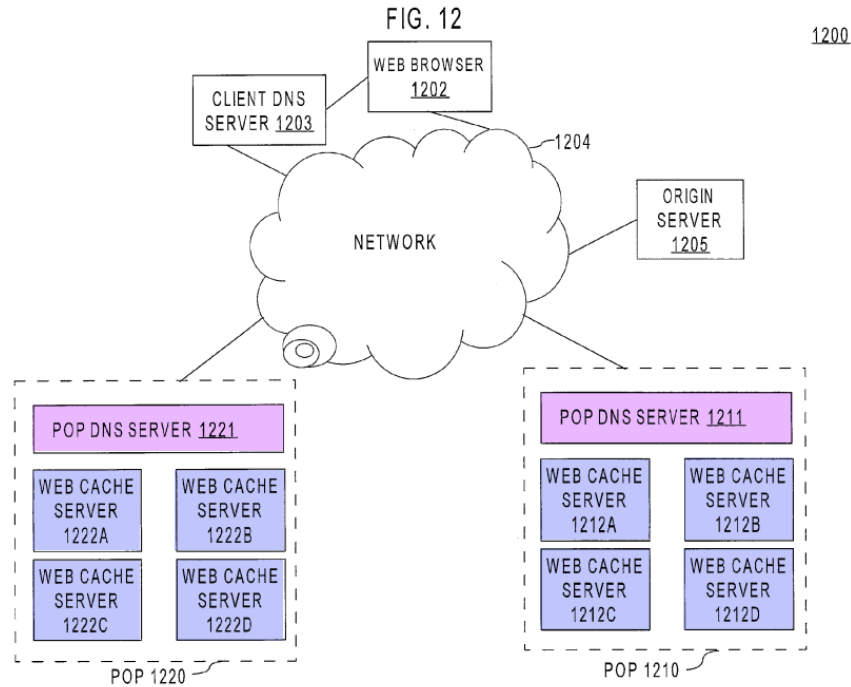
Liu discloses distributing web content to a plurality of web cache servers, or replicating DNS content between DNS cache servers. §VII.B.4 (cls. 2/28); *Lanning*, ¶210.

A POSITA would have understood that, upon joining a CDN (cls. 2/28) and/or after joining the CDN (cl. 28), both content- and DNS-type cache servers would obtain either web content or DNS content from other caches in the CDN to host the content for itself in the first instance or fetch content when receiving a request for content it is not storing (for web-cache servers), or to update the (DNS) content as needed (for DNS-cache servers). §VII.B.4 (cls. 2/28); *Lanning*, ¶211.

10. Claims 12/38

Liu discloses or suggests claims 12/38's additional limitations. *Lanning*, ¶¶212-15.

Liu depicts the UDN includes groups of peer servers (e.g., 1222A-D, and separately 1212A-D) at each POP, as well as a separate tier of peer DNS servers spread across different POPs. *See, e.g., Liu*, Fig. 12; *Lanning*, ¶213. Liu also explains how DNS servers have sets of records “that are replicated on multiple servers” and “transfer zones from each other to maintain the consistency of information in the DNS.” *Liu*, ¶6.



A POSITA would have considered it obvious that, upon joining a CDN (cls. 2/28) and/or after joining the CDN (cls. 28), both content- and DNS-type cache servers would obtain either web content or DNS content from their peer caches in the CDN. §VII.B.4 (cls. 2/28); *Lanning*, ¶214. A POSITA would have understood that because peers are typically more numerous than higher-tier servers and/or “may be closer in terms of network distance,” acquiring data from these peers would be faster than obtaining content from higher-tier servers and “could also lessen the burden” placed on those higher-tier servers. *Lanning*, ¶215 (quoting *Seed*, ¶62).

11. Claims 13/39¹¹

Liu discloses or suggests claims 13/39's additional limitations. *Lanning*, ¶¶216-19.

Liu discloses that the global-configuration information includes “the list of POPs and the servers present at each POP,” a static mapping of domains, servers, and services. *Liu*, ¶¶184, 205, 296-99; §VII.B.3.d-VII.B.3.e (1(b)(1)-(b)(2)/27(A)-(B)). A POSITA would have understood that, upon joining the CDN (cls. 2/28), the server would use the information provided in the configuration file to locate resources—whether additional software or content—stored elsewhere in the network to install/load onto the server. §VII.B.4 (cls. 2/28); *Lanning*, ¶217. Indeed, identifying the locations of resources is in part the purpose of the global-configuration file. *Lanning*, ¶218.

A POSITA would have also understood that, after joining the CDN (cl. 28), the server would also use the global-configuration file to locate resources not already located on the server—such as updated DNS tables or to fetch/re-direct requests for content missing from the server that receives the request. *Liu*, ¶¶184, 414-417, 423, 436; *Lanning*, ¶219.

¹¹ See *supra* n.8.

12. Claims 14/40

Liu discloses or suggests claims 14/40's additional limitations. *Lanning*, ¶¶220-21.

Liu discloses obtaining original and updated “centrally managed” global-configuration files, which a POSITA would have understood would come from a centralized area like Liu's NOC (*obtains said global-configuration data from one or more locations in the CDN*). §§VII.B.3.d-VII.B.3.e (1(b)(1)-1(b)(2)/27(A)-(B)).

13. Claims 15/41

Liu discloses or suggests claims 15/41's additional limitations. *Lanning*, ¶¶222-23.

A POSITA would have understood that the original and updated “centrally managed” global-configuration files would be obtained from the NOC (*control core*). §§VII.B.3.d-VII.B.3.e (1(b)(1)-1(b)(2)/27(A)-(B)).

14. Claims 16/42

Liu discloses or suggests claims 16/42's additional limitations. *Lanning*, ¶¶224-26.

Liu discloses the NOC (*control core*, §VII.B.3.b (1(a)/27(pre)(a))) contains a plurality of machines that each perform different tasks, such as NetView, AdminTools, and LogServer machines:

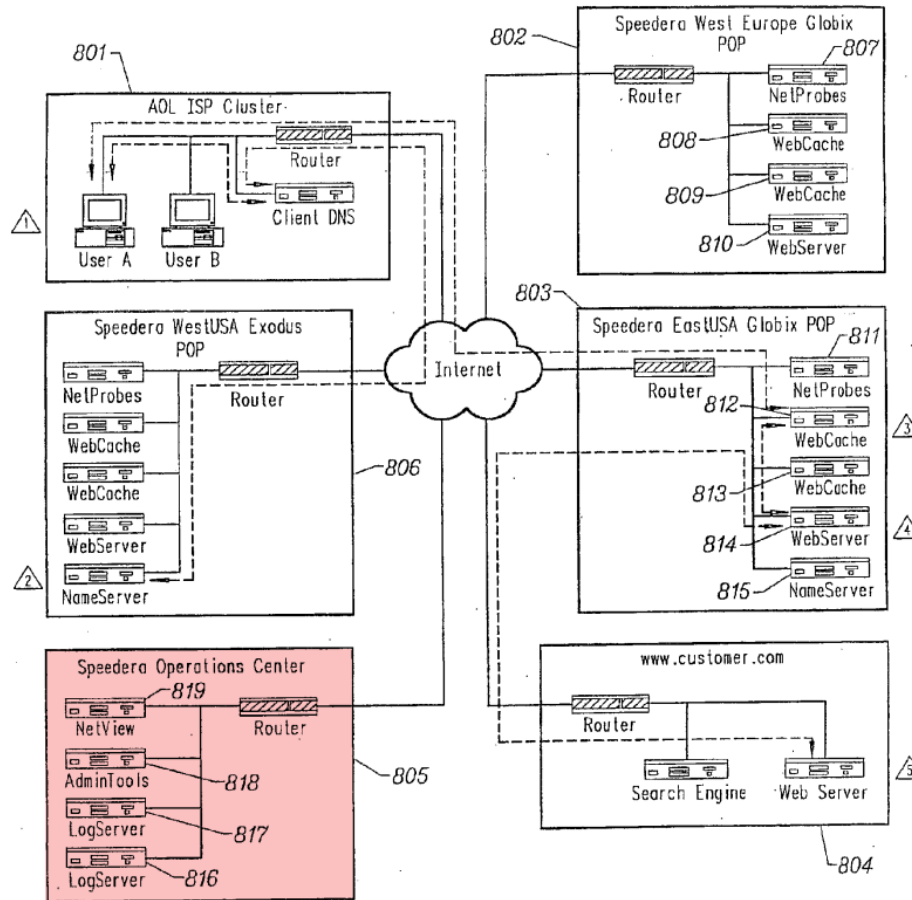


FIG. 8

Liu, Fig. 8, ¶¶17, 263-64; *Lanning*, ¶225. A POSITA would have understood these machines constitute a “distributed system” as claimed because they are distinct machines that operate together as part of a centralized operations center. *Lanning*, ¶226.

15. Claims 18/44

Liu discloses or suggests claims 18/44’s additional limitations. *Lanning*, ¶¶227-30.

Liu discloses that the configuration file maps domain hostnames (*global-configuration data*). *Liu*, ¶221. Liu also uses an algorithm to “**validate** the hostnames in the configuration file” (*determining validity of a version of global-configuration data*), (*id.*, ¶¶223-39), and is capable of “flag[ging] an error” for hostnames in the configuration file (*determine that said version of said global-configuration data...is invalid*). *Id.*, ¶226; *Lanning*, ¶228.

Liu also discloses that DNS cache servers can “periodically query the origin server for updates.” *Liu*, ¶435.

A POSITA would have understood that a (web or DNS) cache server with an invalid hostname would have prompted the cache server to query an update to ensure it has a correct mapping of server hostnames to route traffic correctly. *Liu*, ¶434 (“[E]dgeDNS cache servers can be deployed at any number of POPs[and] configured to serve multiple domains as well as multiple customers”); *Lanning*, ¶230.

16. Claims 19/45

Liu-Devanneaux teaches or suggests claims 19/45. *Lanning*, ¶¶231-33.

Liu discloses that cache servers periodically check the validity of customer-configured content on their servers against what the customer provides on the origin server (*by said particular cache server...determining*). *Liu*, ¶81. Specifically, the cache may send an “If-modified-since” request to the origin site “to see if the content has changed” (*determining validity of a version*). *Id.* If the content has changed, “the

old content may be invalidated and the new content published to all the caches in the network” (*obtains customer-configuration information*). *Id.*, ¶¶81-82. Content may also be invalidated manually by the customer. *Id.*, ¶¶90-96; *Lanning*, ¶232.

A POSITA would have recognized invalidated ***cached content*** as ***invalidated customer-configuration data*** because the status (valid or invalid) of the cached content is controlled by the customer (either by manually invalidating old content, or updating content at the origin server, after which the cache servers update their content). *Lanning*, ¶233; §VII.B.3.f (cls. 1(b)(2)(2)/27(C)); *Nygren*, 12 (describing cache control and indexing as a customer-controlled and metadata-implemented configuration).

17. Claims 20/46

Liu discloses or suggests claims 20/46’s additional limitations. *Lanning*, ¶¶234-36.

Liu discloses that the cache periodically checks the validity of its content. *Liu*, ¶81; §VII.B.16 (cl. 19). Liu suggests such checks are at predetermined time intervals, but notes that “[t]his interval can be configured depending upon ones needs.” *Liu*, ¶81; *Lanning*, ¶235.

A POSITA would have found it obvious to modify Liu by conditioning the cache’s periodic check to be *in response to a request to serve content*, rather than a predetermined time interval. *Lanning*, ¶236. This would conserve network-

processing resources by updating caches only when necessary to respond to requests, rather than perpetually checking for updates that may not exist when content has not changed. *Liu*, ¶81; *Lanning*, ¶236.

18. Claim 21

Liu discloses or suggests claim 21's additional limitations. *Lanning*, ¶¶237-39.

Liu discloses that the UDN comprises a DNS Load Balancer, which “dynamically selects the best suitable edge DNS cache server to resolve a domain name resolution request” (*Liu*, ¶430) by evaluating a number of factors including server traffic load, latency, and the likelihood that a given server has the content cached (*id.*, ¶¶422-24); *Lanning*, ¶238. A POSITA would have understood a DNS to manage content requests constitutes a *rendezvous mechanism* consistent with the '053 patent. *Lanning*, ¶239; '053 patent, 4:29-35 (“the rendezvous system 104 is used to direct client resource requests [and] is preferably implemented using the DNS”).

19. Claims 22/47(pre)

Liu discloses or suggests claims 22/47(pre)'s additional limitations. *Lanning*, ¶¶240-42.

Liu discloses the UDN includes a system that includes “Logging System,” “Log Distributor,” “Log Collector,” “Log Server,” and other associated protocols

(*collector system*), wherein the Log Server (*collector mechanism*) “collects log files from the POP servers [(*at least one cache server*)] via...log collector daemons.” *Liu*, ¶¶56, 363. A POSITA would have understood that the Log Server constitutes a *collector mechanism* consistent with the ’053 patent because it collects the logs from the POP server for further distribution. *Id.*, ¶¶56, 366, 369; ’053 patent, 41:10-14 (“The collector mechanisms...collect information regarding resources delivered on behalf of content providers (e.g., logs and performance data) from the caches.”); *Lanning*, ¶¶241-42.

20. Claims 23/47(E)-(F)

Liu discloses or suggests claims 23/47(E)-(F)’s additional limitations. *Lanning*, ¶¶243-44.

Liu discloses that “[a] POP Server [(*each particular cache server*)] is any server in a POP that runs [(*constructed and adapted*)] a log distributor daemon that sends log files to the log collector daemons on the log servers,” (*provide at least some of said generated information to said collector system*) and that “[f]or each unique customer...the server must create a separate log file” (*generate information*). *Liu*, ¶¶363, 365. Liu then walks through exactly how log files are sent from the servers to the Log Server and how they will be processed thereafter. *Id.*, ¶¶366-90; *Lanning*, ¶244.

21. Claims 24/48

Liu discloses or suggests claims 24/48's additional limitations. *Lanning*, ¶¶245-49.

Liu discloses that logs events are streamed to a file on each cache server, and the file is then transmitted to a Log Server (*collector mechanism*) for further processing. *Liu*, ¶¶370, 377.

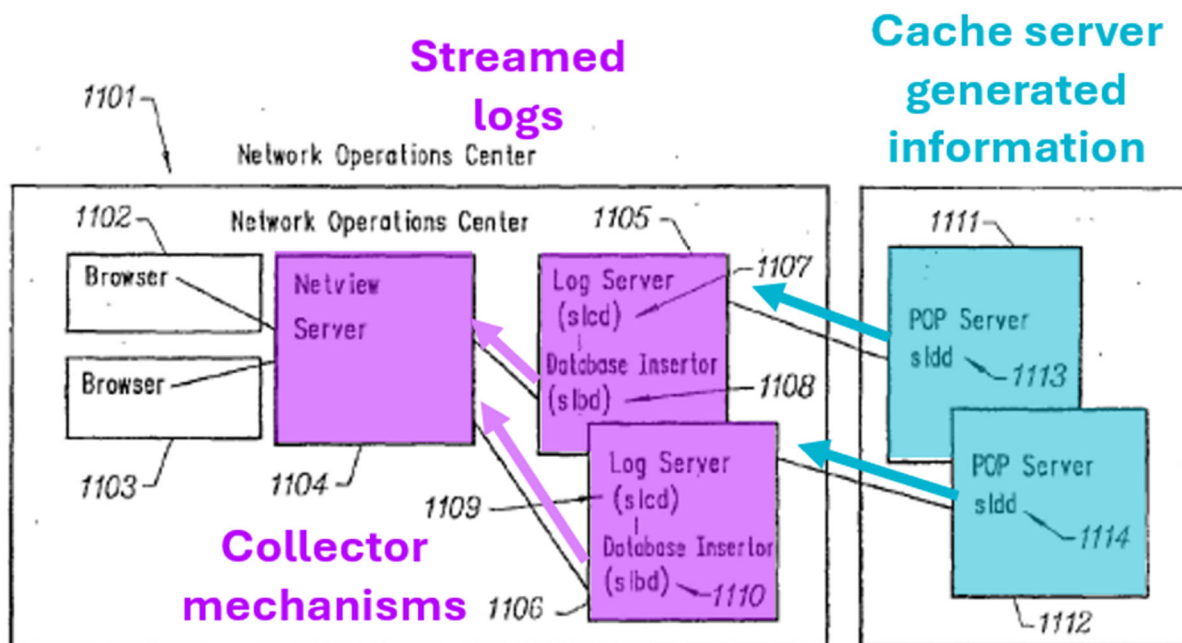


FIG. 11

Id., Fig. 11.

To the extent transmitting intermittent log files does not constitute “streaming,” a POSITA would have understood that there are only two temporal ways to transmit

data from one network element to another: (1) transmit entire files at once; or (2) stream data in real time. *Lanning*, ¶247; *Liu*, ¶55 (noting media may be “a live streaming event” compared to “large file downloads”). A POSITA would have understood that these are two well-known and obvious alternatives; modifying *Liu* to expressly “stream” log files would have been nothing more than applying one known technique (streaming) as a suitable alternative to another (file transfer) to solve a known problem in the art (transfer data from one network element to another) and/or to achieve a predictable result (transferring log data to the Log Server). *Lanning*, ¶248; *see also, e.g., Gibson*, 2:15-18 (“The system may also provide analysis of stream of fault log data or daily download of fault log data”); *McGuire*, ¶48 (“The Log Analyzer 708 may receive the stream of Log Data directly from a web server, from an intermediary (for example, a data storage device), or on a real-time basis”); ’053 patent, 20:15-17 (acknowledging streaming and file transfers are known alternatives); *Intel*, 61 F.4th at 1380 ; *Snap*, 2025 WL 2793538 at *3-4.

Moreover, a POSITA would have been expressly motivated to modify *Liu* to stream log data in real time, rather than transfer compiled log files to ensure the Log Server has real-time data concerning the performance of the edge servers. *Lanning*, ¶249. And because *Liu* itself already knows how to implement streams of (content) data, implementing streaming protocols for simpler log files would have been well within the skills of a POSITA. *Id.*

22. Claim 49

Liu-Devanneaux teaches claim 49. *Lanning*, ¶¶250-51.

When implementing a CDN that utilizes customer-specific configuration information (Liu) or files (Devanneaux), a POSITA would have understood that the CDN would receive configuration information from multiple customers whose content is served by each respective cache, and that content would be served according to the global-configuration data (applied to all customers), and the customer-specific configuration provided by the customer whose content is being served. §§VII.B.3.d-VII.B.3.g (1(b)(1)-1(b)(2)(3)/27(A)-27(D)); *Lanning*, ¶251.

23. Claims 25/50

Liu-Devanneaux teaches or suggests claims 25/50. *Lanning*, ¶252.

As described above, the Liu-Devanneaux CDN caches serve content according to *global* and *customer configuration information* that is associated with the CDN and stored in copies of *configuration* files on each cache in the CDN. §VII.B.3.f-VII.B.3.g (cls. 1(b)(2)(2)-(3)/27(C)-(D)). *Lanning*, ¶252.

24. Claim 26

Liu-Devanneaux teaches or suggests claim 26. *Lanning*, ¶¶253-54.

The '053 patent discloses that an executive system (or “Executive”) “control[s] all activities within the cache” with the “responsibility...to maintain a prioritized list of runnable tasks...in a priority order.” ’053 patent, 23:45-50. Liu’s POPs (*plurality of cache servers*) have servers with “a mix of Speedera software” (*executive system*)

including e.g., WebCache, NameServer, and NetProbes, each of which perform several tasks concurrently. *Liu*, ¶¶44, 248-53, 307-13 (NetProbe), 245, 305 (WebCache), 244 (NameServer). *Liu* also explains that “[a]ny combination of the software components...can be installed on a single machine.” *Id.*, ¶259. For instance, if an edge server is configured as both a WebCache and NetProbe, that server may, concurrently, respond to requests for content while monitoring performance. *Id.*, ¶¶245, 305 (WebCache), 313 (noting NetProbe “probes run constantly and send results to servers running NameServer software.”). Also, NetProbes include latency probes (*id.*, ¶ 252) running constantly (*id.*, ¶254) and are **prioritized** according to accuracy (*id.*, ¶49); *Lanning*, ¶254.

C. Ground 2B: Obviousness over Liu, Devanneaux, and Nygren (Claims 17, 43)

1. Prior Art

a. Nygren

Nygren provides an overview of the well-known Akamai CDN. *Nygren*, 2. For instance, Nygren describes the structure of the Akamai CDN which involves origin servers, edge servers, and control systems (like *Liu*), *id.*, 5-7; *Lanning*, ¶255, and explains how the network requires “[c]onfiguration changes and software updates” that need to be rolled out efficiently. *Nygren*, 6; *see also id.* at 14-15.

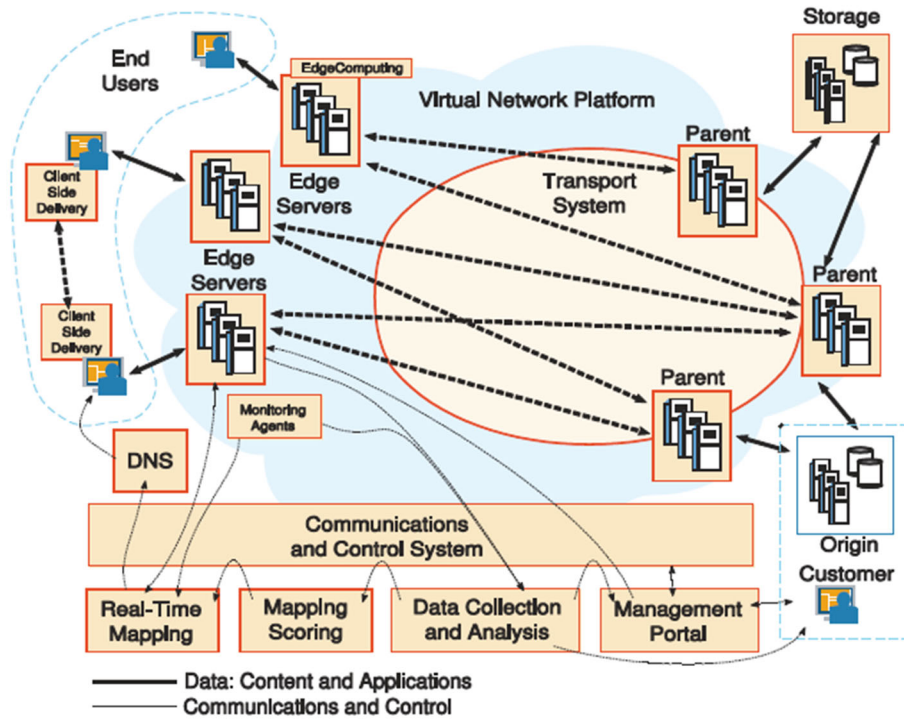


Figure 5: System components of the Akamai platform.

Id., Fig. 5, 12-14; *Lanning*, ¶¶256-58.

2. Claims 17/43

Liu-Devanneaux-Nygren teaches or suggests claims 17/43. *Lanning*, ¶¶259-63.

Liu discloses that NOCs contain a plurality of hardware elements that each serve different functions, and that multiple NOCs may be established for redundancy. *Liu*, ¶¶262-264, Fig. 8. However, Liu does not expressly disclose how the network resolves any discrepancies between system elements in the NOC, or between different NOCs. *Lanning*, ¶260.

Nygren, however, taught a protocol called “PAXOS” and “decentralized leader election” (*consensus algorithms*), which Akamia’s CDN used for fault

tolerance “to accommodate for the possibility of failed system components,” as well as for “key distribution servers” to coordinate key replication. *Nygren*, 6, 14.

When building a system with multiple central commands (NOCs) like Liu’s, a POSITA would have been motivated to implement protocols like PAXOS to determine uniform control commands based on consensus of the various machines in each NOC (*wherein the control core uses a distributed consensus algorithm to achieve consensus among the plurality of machines*). *Lanning*, ¶262. Indeed, the ’053 patent itself envisioned using PAXOS as its control-core consensus algorithms for this exact purpose. *’053 patent*, 18:40-20:13.

A POSITA would have had a reasonable expectation of success in making this modification because Nygren describes implementing PAXOS protocols in a content-delivery network like Liu’s. *Lanning*, ¶263. Moreover, the ’053 patent itself acknowledged that numerous prior-art and commercially available PAXOS protocols were known and implemented in distributed networking systems. *’053 patent*, 19:42-20:9; *Lanning*, ¶263. Thus, implementing commercially available protocols like PAXOS for their intended purpose would have been well within the skills of a POSITA. *Lanning*, ¶263.

D. Ground 2C: Obviousness over Liu, Devanneaux, and Lewin819 (Claims 3, 5, 7-8, 29, 31, 33-34)

1. Prior Art

a. Lewin819

Lewin819 is another reference that discloses implementation details of Akamai's CDN. *Lewin819*, Cover (assigned to Akamai), Abstract (“disclosed technique...on content delivery network (CDN) edge servers”). Specifically, Lewin819 describes the ability to configure CDN edge servers to build dynamic webpages on behalf of customers using scripts defined by content providers (e.g., customers), rather than having websites assembled by some central application server. *Id.*, Abstract, ¶¶12-13, 35-36. For instance, Lewin819 depicts one exemplary website template in which scripts control what content is filled into blocks (1), (2), and (4):

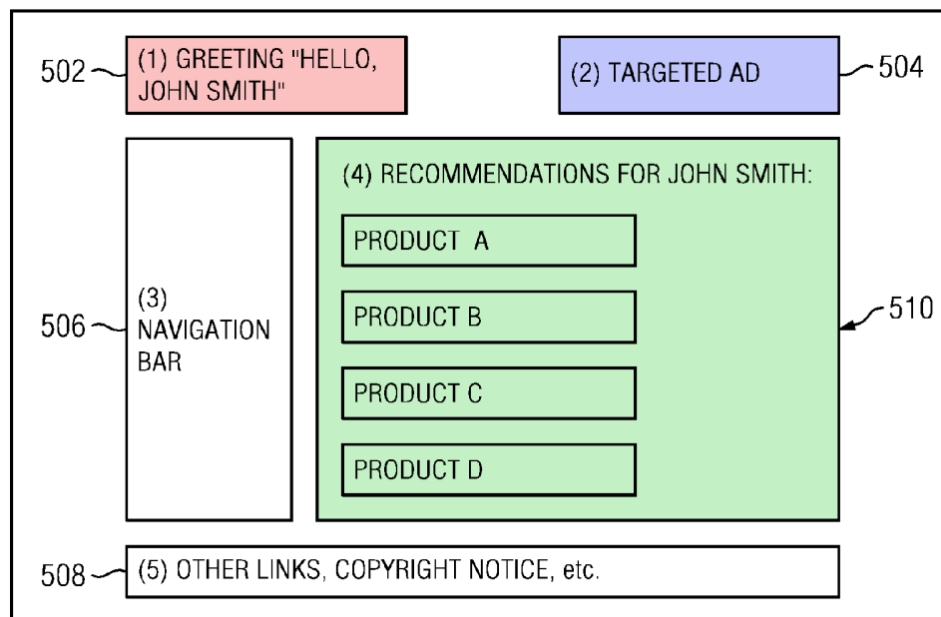


FIG. 5

Id., Fig. 5. A POSITA would have recognized that the associated scripts shown below as an html file that contains multiple ESI statements (“<esi.”) that are “used to define Web page fragments for dynamic assembly at the edge.” *Id.* Abstract.

```
<html>
<body>
<!--personalized greeting (1)-->
Hello ${HTTP_COOKIE{"username"}}

<!--targeted ad (2)-->
<esi:choose>
<esi:when test="${GEO{country_code})='US' ">
    <esi:include src=us_ad.html/>
</esi:when>
<esi:when test="${GEO{country_code})='Canada' ">
    <esi:include src=canada_ad.html/>
</esi:when>
<esi:otherwise>
    <esi:include src=generic_ad.html/>
</esi:otherwise>
</esi:choose>

<!--Static navigation bar (3)-->
<a href=...> <a href=...> <a href=...> <a href=...>

<!--Personalized recommendations (4)-->
<esi:include src=recommendations.html/>

<!--Static links, copyright,etc (5)-->
<a href=...> <a href=...> <a href=...> <a href=...>
Copyright 2001, etc.
</body>
</html>
```

FIG. 6

```
<esi:include src="products/A.html"/>
<esi:include src="products/B.html"/>
<esi:include src="products/C.html"/>
<esi:include src="products/D.html"/>
```

FIG. 7

Id., Figs. 6-7, ¶¶38-43; *Lanning*, ¶264.

2. Motivation to Combine with Lewin819

Liu discloses that “[c]ustomers [(e.g., content providers)] can leverage the size, scope, and location of the UDN to store content such as HTML, images, video, sound and software for fast and highly available access by clients.” *Liu*, ¶29; *see also id.*,

¶241. Although Liu does not expressly describe using CDNs to serve dynamic webpages, those in the art had appreciated that CDNs—which were initially limited to loading static content on websites—had (prior to 2011) expanded to “accelerate entire web or IP-based applications.” *See, e.g. Nygren*, 4. With such expanded use cases came a greater need to provide content owners options to configure the delivery of their content, including dynamic webpages and applications. *Lanning*, ¶265.

A POSITA would have been motivated to modify Liu’s UDN to further include the ability to serve dynamic webpages that the UDN would construct using customer-configurable scripts, such as those taught by Lewin819. §VII.D.1.a (Lewin819 Summary); *Lanning*, ¶266. As Lewin819 explains, this would have provided customers with additional features to build dynamic websites without burdening their own web/application servers to build these websites, because the CDN would do it for them, while also doing so at the edge of the network to ensure websites were built quickly. *Lewin819*, ¶¶9, 12-13; *Lanning*, ¶266.

A POSITA would have had a reasonable expectation of success in making this modification because Liu already envisions serving website content and software. *Liu*, ¶¶29, 241; the only difference is that the edge servers would be configured to receive web templates and scripts from customers to load dynamic content elements rather than just serving static content to clients or the customer’s web/application

servers. *Lanning*, ¶267. Moreover, Lewin⁸¹⁹ provides the necessary framework to carry out these dynamic webpages and scripts in a CDN environment like Liu's. *Lewin*⁸¹⁹, ¶¶34-36; *Lanning*, ¶267 (citing the ESI Language Specification 1.0¹² and explaining that ESI logic uses ESI scripts and metadata tags to assemble and deliver cached content).

3. Claims 3/29

Liu-Devanneaux-Lewin⁸¹⁹ teaches or suggests claims 3/29. *Lanning*, ¶¶268-70.

Liu and Devanneaux both teach customer-configuration information/files that are programmed by the customer and then distributed to servers in the CDN. §VII.B.3.g (Ground 2A, 1(b)(2)(2)/27(C)); *Lanning*, ¶269. However, neither Liu nor Devanneaux expressly discuss a server obtaining customer-configuration information upon receipt of a request for a resource associated with the customer. *Lanning*, ¶269.

Upon implementing a CDN capable of building dynamic webpages using customer-provided metadata, §VII.D.2 (motivation), a POSITA would have

¹² See *ESI Language Specification 1.0* by Mark Tsimelzon of Akamai Technologies, et al., August 4, 2001, available at <<http://www.w3.org/TR/2001/NOTE-esi-lang-20010804>>.

understood that Liu's servers would obtain the customer-configuration information (including but not limited to scripts) in response to a request for content (e.g., a dynamic web page). *Lanning*, ¶270. For instance, Lewin819 teaches that customer-configuration information rules may come in the form of metadata, which may include "HTTP response headers, in a configuration file, *or a request itself*." *Lewin819*, ¶36. Lewin819 also teaches that "a given request received at the edge server is mapped to a given base processor, preferably by content provider-specific metadata...to create an assembled response that is then sent back to the requesting end user." *Id.*, ¶14. In other words, the server would obtain details on how to implement dynamic webpages from the metadata and/or scripts that may be provided in the request itself, or as a result of receiving a request when building the response. *Lanning*, ¶270.

4. Claims 5/31

Liu-Devanneaux-Lewin819 teaches or suggests claims 5/31. *Lanning*, ¶¶271-74.

Liu and Devanneaux both teach customer-configuration information/files that are programmed by the customer and then distributed to servers in the CDN. §VII.B.3.f (Ground 2A, 1(b)(2)(2)/27(C)); *Lanning*, ¶272. Devanneaux also teaches the ability to at least **parse** configuration information from metadata files by reading scripts. *Devanneaux*, Fig. 5, ¶¶18, 28-29; *Lanning*, ¶272. However, neither Liu nor

Devanneaux expressly discuss a server in which the customer-configuration information itself is a script. *Lanning*, ¶272.

Upon implementing a CDN capable of building dynamic webpages using customer-provided metadata, §VII.D.2 (motivation), a POSITA would have understood that Liu's servers would be configured to receive customer-configuration information that includes scripts in order to build the dynamic websites. *Id.*; *Lanning*, ¶273. A POSITA would have understood that Lewin819, for instance, uses scripts to enable the delivery of **dynamic** content, as opposed to the same static content in response to each request. *Lanning*, ¶273.

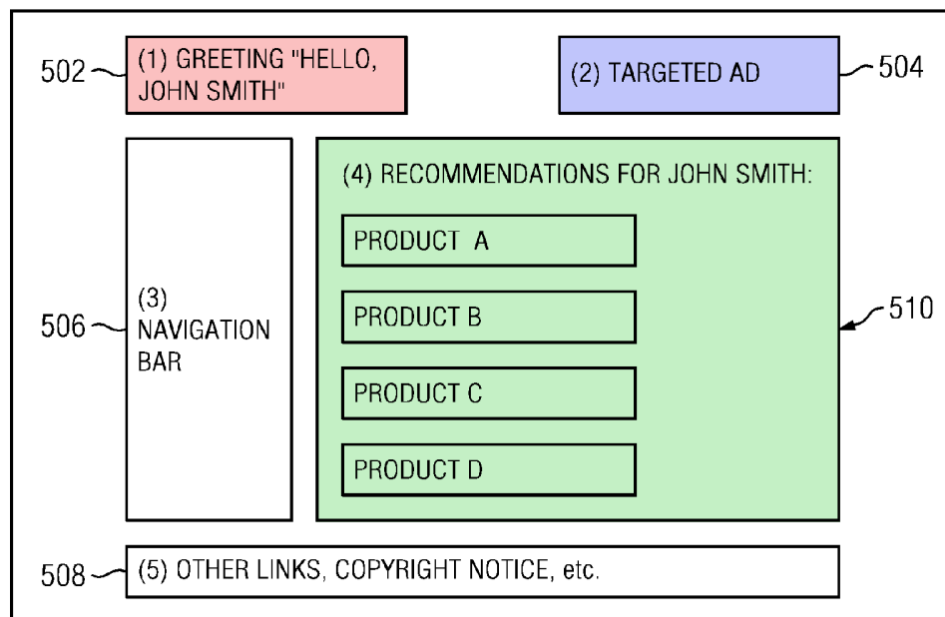


FIG. 5

```
<html>
<body>
<!--personalized greeting (1)-->
Hello ${HTTP_COOKIE{"username"}}

<!--targeted ad (2)-->
<esi:choose>
<esi:when test="${GEO{country_code})='US' ">
    <esi:include src=us_ad.html/>
</esi:when>
<esi:when test="${GEO{country_code})='Canada' ">
    <esi:include src=canada_ad.html/>
</esi:when>
<esi:otherwise>
    <esi:include src=generic_ad.html/>
</esi:otherwise>
</esi:choose>

<!--Static navigation bar (3)-->
<a href=...> <a href=...> <a href=...> <a href=...>

<!--Personalized recommendations (4)-->
<esi:include src=recommendations.html/>

<!--Static links, copyright,etc (5)-->
<a href=...> <a href=...> <a href=...> <a href=...>
Copyright 2001, etc.
</body>
</html>
```

FIG. 6

```
<esi:include src="products/A.html"/>
<esi:include src="products/B.html"/>
<esi:include src="products/C.html"/>
<esi:include src="products/D.html"/>
```

FIG. 7

Lewin819, Figs. 5-7, ¶¶12-13, 35-36, 38-43.

Moreover, providing scripts in customer-configuration information was well-known for a range of purposes other than just delivering dynamic content, such as providing customers the ability to set custom response rules that require independent processing to carry out. *Lanning*, ¶274 (discussing *Cuan* ¶26, describing use of

scripts in configuration files to enable custom processes, such as notifying customers of successful content deliveries).

5. Claims 7/33

Liu-Devanneaux-Lewin819 teaches or suggests claims 7/33—that the customer configuration information comprises customer-provided scripts used by the CDN to build dynamic websites (*customer-configuration script*). §VII.D.4 (cls. 5/31); *Lanning*, ¶275.

6. Claims 8/34

Liu-Devanneaux-Lewin819 teaches or suggests claims 8/34. *Lanning*, ¶¶276-77.

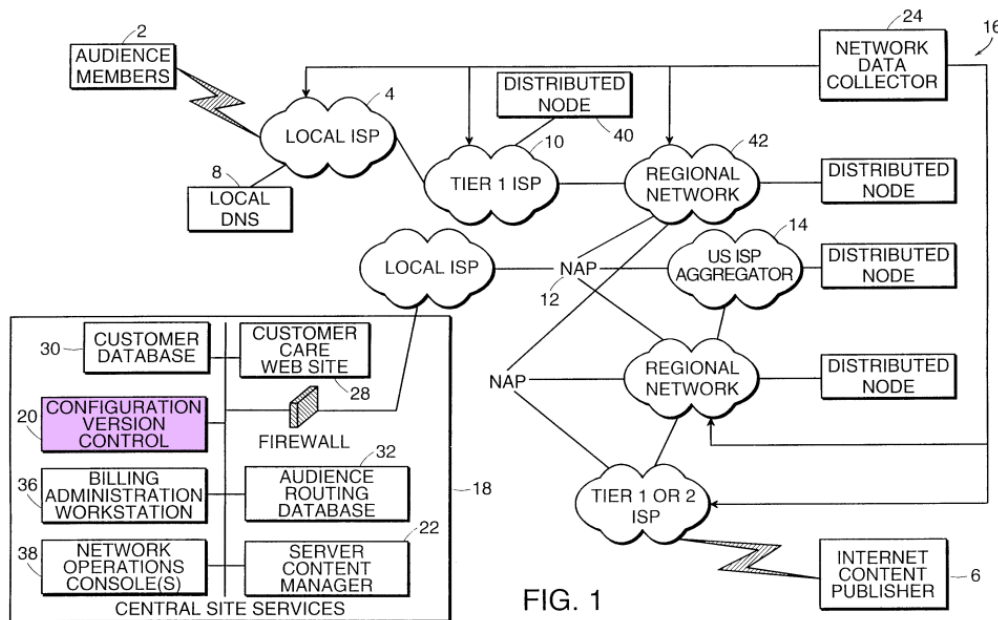
As mentioned, (§VII.D.3 (cls. 3/29)), not only does Lewin819 describe using customer-configuration scripts to serve dynamic webpages from a CDN, but Lewin819 also notes that customer-configuration metadata may specify processing requirements for the CDN. *Lewin819*, ¶14 (noting “content provider-specific metadata” may specify mapping requests to processor instantiations to compile a complete response); *Lanning*, ¶277.

E. Ground 2D: Obviousness over Liu, Devanneaux, and Crosbie (Claims 18-20, 44-46)

1. Prior Art

a. Crosbie

Crosbie discloses a CDN that uses configuration files to dictate how content is cached and mirrored throughout the network. *Crosbie*, Abstract, 6:1-6. Crosbie provides a “Configuration Version Control System 20 [that] serves as the central point of control and audit for server configuration files throughout the system,” which enables “full roll-forward and rollback functionality” to control the release of new configuration files, or reverting back to prior configuration files. *Id.*, 10:31-11:9, Fig. 1; *Lanning*, ¶278.



2. Motivation to Combine with Crosbie

Liu notes that “[a]t any time, a new configuration file can be pushed to all machines that need it.” *Liu*, ¶301. However, Liu does not expressly discuss how network elements engage in version control to ensure they are not using outdated versions of their configuration files that may produce errors. *Lanning*, ¶279.

A POSITA implementing Liu's UDN would have been motivated to look to other art to implement a version-control scheme to ensure that the UDN can roll out new configuration files or roll back to prior configurations as needed. *Id.*, ¶280. Crosbie, for instance, provides one such version control-scheme. *See* §VII.E.1.a (Crosbie).

A POSITA would have been motivated, for instance, to implement Configuration Version Control System software instructions like Crosbie's in Liu's network to keep track of different versions of the configuration files Liu (and Devanneaux) teaches, and ensure servers have the most-recent version when making service or configuration decisions, providing a "simplifie[d] change tracking, auditability," and "rollback" process for configuration files. *Crosbie*, 10:31-11:9; *see also Liu*, Fig. 8 (805), ¶¶184, 205-206, 243-59; *Lanning*, ¶280. Incorporating a version-control system like Crosbie's would have been nothing more than applying a known technique (version control) to improve a prior-art device (a CDN network with updatable configuration files) to solve a known problem or achieve a predictable result (providing mechanisms to ensure version consistency and to roll out/roll back updated versions of configuration information/files). *Lanning*, ¶280.

A POSITA would have a reasonable expectation of success in making this modification because the modification is straightforward; Liu and Devanneaux already teach using customer-configuration information/files, and all Crosbie

provides is a centralized system to track the versions of those configurations used throughout the network. *Id.*, ¶281. The minimal coding necessary to implement centralized version control (such as configuring a central database to track configuration-information versions), and to have servers check their versions would have been well within the skills of a POSITA. *Id.*

3. Claims 18/44

Liu-Devanneaux-Crosbie teaches or suggests claims 18/44; *Lanning*, ¶¶282-84.

Liu discloses cache servers operate in accordance with global configuration files that can be updated as needed. §VII.B.3.e (1(b)(2)-1(b)(2)(1)/27(B)); *Lanning*, ¶283.

Upon implementing a centralized version control module in Liu's UDN, §VII.E.2 (motivation), a POSITA would have understood that Liu's cache servers would be configured to evaluate which version of the global configuration file they possess (*determine validity of a version of global configuration data*) by comparing their version to the one in the centralized module. *Lanning*, ¶284. Such version-checking mechanisms were extremely old and well-known in the art. *Id.* (citing *Lee*, ¶51 (checking central server to determine whether updated software module available), *Lee* 617, ¶¶13-14 (describing version checks for software updates), *Liskov*, 7:29-43 (describing CDN management server checking a version of a catalog)). If

that comparison revealed an outdated version, a POSITA would have understood that the cache would be configured to obtain the most-recent version as indicated in the version control module (*obtains updated global configuration data...when said particular cache server determines that said version of said global configuration data stored on said particular cache server is invalid*). *Lanning*, ¶284; *Crosbie*, 10:31-11:9.

4. Claims 19/45

Liu-Devanneaux-Crosbie teaches or suggests claims 19/45; *Lanning*, ¶¶285-87.

Liu, alone or combined with Devanneaux, teaches cache servers operate in accordance with customer-configuration information (or files based on Devanneaux) that can be updated as needed. §VII.B.3.f (1(b)(2)(2)/27(C)); *Lanning*, ¶286.

Upon implementing a centralized version control module in Liu's UDN, §VII.E.2 (motivation), a POSITA would have understood that Liu's cache servers would be configured to evaluate which version of the customer-configuration information/file they possess (*determine validity of a version of particular customer configuration information*) by comparing their version to the one in the centralized module. *Lanning*, ¶287. Such version-checking mechanisms were extremely old and well-known in the art. *Id.* (citing *Lee*, ¶51, *Lee617*, ¶¶13-14, *Liskov*, 7:29-43). If that comparison revealed an outdated version, a POSITA would have understood that the

cache would be configured to obtain the most-recent version as indicated in the version control module (*obtains customer-configuration information...when said particular cache server determines that said version of said particular customer configuration information stored on said particular cache server is invalid*). *Id.*, ¶287; *Crosbie*, 10:31-11:9.

5. Claims 20/46

Liu-Devanneaux-Crosbie teaches or suggests claims 20/46; *Lanning*, ¶¶288-290.

A POSITA would have appreciated that a UDN like the one in the combined system of Liu-Devanneaux-Crosbie, with its centralized version-control system, would check for updates to configuration information in one or more ways: (1) manually in response to a push from a customer (as Liu envisions); (2) automatically at predetermined time intervals or specified times; or (3) in response to a triggering event, such as a request for content. *Lanning*, ¶289. Choosing any of these options would have been straightforward, with each having certain design tradeoffs. *Id.* But a POSITA would have specifically chosen to check the version of customer-configuration information in response to a request to ensure that the most-recent version of customer-configuration information was used in servicing the request. *Id.*, ¶290 (*citing Davis*, Fig. 11, 14:18-19).

VIII. CONCLUSION

The Challenged Claims are unpatentable and institution of IPR is respectfully requested. *Lanning*, ¶¶291-92.

IX. MANDATORY NOTICES

A. Real Party in Interest

The Petitioner and real party-in-interest is Microsoft Corporation.

B. Related Matters

Sandpiper asserted the '053 patent in the following litigation:

- *Sandpiper CDN, LLC v. Microsoft Corp.*, No. 2-25-cv-00664
(E.D. Tex., June 26, 2025)

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

LEAD COUNSEL
Jessica Kaiser (Reg. No. 58,937) kaiser-ptab@perkinscoie.com PERKINS COIE LLP 1900 Sixteenth Street, Suite 1400 Denver, CO 80202-5255 Telephone: (303) 291-2300
BACK-UP COUNSEL
Christopher Marando (Reg. No. 67,898) Marando-ptab@perkinscoie.com PERKINS COIE LLP 700 Thirteenth Street, N.W. Suite 800 Washington, DC 20005-3960 Telephone: (202) 654-6227 Steven Beigelmacher (Reg. No. 75,050) beigelmacher-ptab@perkinscoie.com

PERKINS COIE LLP
1301 Second Avenue, Suite 4200
Seattle, WA 98101-3804
Telephone: (206) 359-8000

Sarah Fowler (Reg. No. 54,854)
stahnke-ptab@perkinscoie.com
PERKINS COIE LLP
3150 Porter Drive
Palo Alto, CA 94304
Telephone: (650) 838-4300

Matthew A. Lembo (Reg. No. 75,633)
Lembo-ptab@perkinscoie.com
PERKINS COIE LLP
1155 Avenue of the Americas 22nd Floor
New York, NY 10036-2711
Telephone: (332) 238-2757

Tyler Knox (Reg. No. 79,948)
knox-ptab@perkinscoie.com
PERKINS COIE LLP
1900 Sixteenth Street, Suite 1400
Denver, CO 80202-5255
Telephone: (303) 454-2911

Petitioner consents to electronic service at the following email address:

Microsoft-Sandpiper-IPR@perkinscoie.com.

D. Power of Attorney

A power of attorney is filed herewith according to 37 C.F.R. §42.10(b).

Respectfully submitted,

/ Jessica Kaiser /

Jessica Kaiser
Reg. No. 58,937
Attorney for Petitioner

PERKINS COIE LLP
1900 Sixteenth Street, Suite 1400
Denver, CO 80202-5255

Date: December 29, 2025

CERTIFICATE OF WORD COUNT UNDER 37 CFR §42.24(d)

Pursuant to 37 C.F.R. §42.24(a), Petitioner hereby certifies that portions of the above-captioned Petition for *inter partes* review of U.S. Patent No. 9,456,053, in accordance with and reliance on the word count provided by the word-processing system used to prepare this Petition, that the number of words in this paper is 13,494. Pursuant to 37 C.F.R. §42.24(a), this word count is in compliance and excludes the table of contents, table of authorities, mandatory notices under §42.8, certificate of service, certificate of word count, appendix of exhibits, and any claim listing. This word count was prepared using Microsoft Word.

Respectfully submitted,

/ Jessica Kaiser /

Jessica Kaiser
Reg. No. 58,937
Attorney for Petitioner

Date: December 29, 2025
PERKINS COIE LLP
1900 Sixteenth Street, Suite 1400
Denver, CO 80202-5255

CERTIFICATE OF SERVICE

The undersigned hereby certifies that true copies of the Petition for *inter partes* review of U.S. Patent No. 9,456,053 and supporting materials (Exhibits and Power of Attorney) were served via overnight delivery on the Patent Owner at the correspondence address of record as listed on PAIR:

STERNE, KESSLER, GOLDSTEIN & FOX P.L.L.C.
1101 K Street, NW
10th Floor
WASHINGTON, DC 20005

A courtesy copy was also sent via electronic mail to Patent Owner's litigation counsel listed below:

Andrea Leigh Fair - andrea@millerfairhenry.com
Carolyn E Drell - cdrell@bsfllp.com
Garrett C. Parish - garrett@millerfairhenry.com
Mark D. Schafer - mschafer@bsfllp.com
Philip Eckert - peckert@bsfllp.com
Ryan D. Dykal - rdykal@bsfllp.com

Respectfully submitted,

/ Jessica Kaiser /

Jessica Kaiser
Reg. No. 58,937
Attorney for Petitioner

Date: December 29, 2025