

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

DECLARATION OF MARK R. LANNING

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further, that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Date: 12/24/2025

By: Mark R. Lanning
Mark R. Lanning

TABLE OF CONTENTS

I.	BACKGROUND AND QUALIFICATIONS	1
A.	Educational Background	1
B.	Professional Background.....	2
II.	MATERIALS CONSIDERED	5
III.	LEGAL STANDARDS	5
IV.	TECHNOLOGY BACKGROUND.....	11
A.	The World Wide Web and the Internet	11
1.	Domain Names and URLs	11
2.	Web Request and Response	12
3.	Data Resources/Objects	13
4.	The HTTP and HTTPS Protocols	13
5.	Domain Name Servers and Redirection.....	16
B.	Content Delivery Networks.....	17
V.	'053 PATENT.....	24
A.	Specification.....	24
B.	Prosecution History	27
VI.	LEVEL OF ORDINARY SKILL	28
VII.	CLAIM CONSTRUCTION	29
VIII.	SUMMARY OF OPINIONS.....	29
IX.	DETAILED EXPLANATION OF GROUNDS.....	31
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)	31
1.	Prior Art	31
2.	Motivation to Combine Lewin with Wein.....	33

EX-1003, PTAB Case No. IPR2026-00190
Declaration of Mark Lanning in support of Petition for
Inter Partes Review of U.S. Patent No. 9,456,053 B2

3.	Claims 1, 27	35
4.	Claim 2	49
5.	Claim 3	54
6.	Claim 4	55
7.	Claim 6	56
8.	Claim 8	57
9.	Claim 9	58
10.	Claim 11	59
11.	Claim 12	60
12.	Claim 13	61
13.	Claim 14	62
14.	Claim 15	64
15.	Claim 16	64
16.	Claim 21	66
17.	Claim 22	67
18.	Claim 23	68
19.	Claim 49(pre)	69
20.	Claim 25	70
B.	Ground 2A: Obviousness over Liu and Devanneaux (Claims 1-2, 4, 6, 9-16, 18-28, 30, 32, 35-42, 44-50)	70
1.	Prior Art	70
2.	Motivation to Combine Liu with Devanneaux	73
3.	Claims 1, 27	77
4.	Claim 2	89
5.	Claim 4	92
6.	Claim 6	93
7.	Claim 9	94
8.	Claim 10	96

EX-1003, PTAB Case No. IPR2026-00190
Declaration of Mark Lanning in support of Petition for
Inter Partes Review of U.S. Patent No. 9,456,053 B2

9.	Claim 11	98
10.	Claim 12	99
11.	Claim 13	101
12.	Claim 14	102
13.	Claim 15	103
14.	Claim 16	103
15.	Claim 18	105
16.	Claim 19	107
17.	Claim 20	108
18.	Claim 21	109
19.	Claim 22	110
20.	Claim 23	111
21.	Claim 24	112
22.	Claim 49	115
23.	Claim 25	116
24.	Claim 26	116
C.	Ground 2B: Obviousness over Liu, Devanneaux, and Nygren (Claims 17, 43)	117
1.	Prior Art	117
2.	Claim 17	120
D.	Ground 2C: Obviousness over Liu, Devanneaux, and Lewin819 (Claims 3, 5, 7-8, 29, 31, 33-34)	122
1.	Prior Art	122
2.	Motivation to Combine with Lewin819	124
3.	Claim 3	126
4.	Claim 5	128
5.	Claim 7	131
6.	Claim 8	132

EX-1003, PTAB Case No. IPR2026-00190
Declaration of Mark Lanning in support of Petition for
Inter Partes Review of U.S. Patent No. 9,456,053 B2

E.	Ground 2D: Obviousness over Liu, Devanneaux, and Crosbie (Claims 18-20, 44-46).....	132
1.	Prior Art	132
2.	Motivation to Combine with Crosbie	133
3.	Claim 18	135
4.	Claim 19	137
5.	Claim 20	139
X.	SUMMARY AND ADDITIONAL REMARKS.....	140

EX-1003, PTAB Case No. IPR2026-00190
Declaration of Mark Lanning in support of Petition for
Inter Partes Review of U.S. Patent No. 9,456,053 B2

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 (“ <i>Lewin</i> ”)
1006	U.S. Patent 7,240,100 (“ <i>Wein</i> ”)
1007	U.S. Publication 2010/0257258 (“ <i>Liu</i> ”)
1008	U.S. Publication 2007/0156845 (“ <i>Devanneaux</i> ”)
1009	Erik Nygren, et al., <i>The Akamai Network: A Platform for High-Performance Internet Applications</i> , 2010 (“ <i>Nygren</i> ”)
1010	U.S. Publication 20100274819 (“ <i>Lewin819</i> ”)
1011	International Application Publication WO 001/13456 (“ <i>Crosbie</i> ”)
1012	U.S. Publication 2002/0065899 (“ <i>Smith</i> ”)
1013	U.S. Publication 2011/0082982 (“ <i>Harvell</i> ”)
1014	U.S. Patent 8,180,720 (“ <i>Kovacs</i> ”)
1015	U.S. Publication 2013/0103556 (“ <i>Lyon</i> ”)
1016	U.S. Patent 7,127,713 (“ <i>Davis</i> ”)
1017	Jennifer Mears, <i>Buyout ends Akamai, Speedera feud</i> (“ <i>Mears</i> ”), March 21, 2005 (“ <i>Mears</i> ”)

EX-1003, PTAB Case No. IPR2026-00190
Declaration of Mark Lanning in support of Petition for
Inter Partes Review of U.S. Patent No. 9,456,053 B2

No.	Exhibit
1018	U.S. Patent 6,343,236 (“ <i>Gibson</i> ”)
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 7340510 (“ <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	(Reserved)
1034	(Reserved)

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.

EX-1003, PTAB Case No. IPR2026-00190
Declaration of Mark Lanning in support of Petition for
Inter Partes Review of U.S. Patent No. 9,456,053 B2

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.

EX-1003, PTAB Case No. IPR2026-00190
Declaration of Mark Lanning in support of Petition for
Inter Partes Review of U.S. Patent No. 9,456,053 B2

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.

EX-1003, PTAB Case No. IPR2026-00190
Declaration of Mark Lanning in support of Petition for
Inter Partes Review of U.S. Patent No. 9,456,053 B2

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)

EX-1003, PTAB Case No. IPR2026-00190
Declaration of Mark Lanning in support of Petition for
Inter Partes Review of U.S. Patent No. 9,456,053 B2

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.

EX-1003, PTAB Case No. IPR2026-00190
Declaration of Mark Lanning in support of Petition for
Inter Partes Review of U.S. Patent No. 9,456,053 B2

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.

I, Mark R. Lanning, do hereby declare:

I. BACKGROUND AND QUALIFICATIONS

1. I have been retained by Petitioner Microsoft Corporation (“Microsoft” or “Petitioner”) to provide my opinions regarding claims 1-50 of U.S. Patent No. 9,456,053 (“the ’053 patent,” Ex-1001).

2. I am currently the President of Telecom Architects, Inc., and Reticle Consulting, LLC.

3. Available at Ex-1004 is a copy of my current curriculum vitae, which provides further details about my academic and professional background and qualifications.

4. I am being compensated for my services and reimbursed for reasonable expenses that I may incur while working on this matter. I have no financial interest in the outcome of this matter or any other matter that may exist between Petitioners and Sandpiper LLC (“Sandpiper” or “Patent Owner”). This compensation does not depend in any way on the conclusions I reach.

A. Educational Background

5. I received a B.S. in Computer Science from Southern Methodist University (SMU) in 1983.

B. Professional Background

6. I have over 40 years of experience working in the telecommunications industry that began in the U.S. Army Signal Corp.

7. In 1983, Digital Switch Corporation (DSC), now a part of Alcatel, hired me, where I was a software development manager on the team responsible for updating DSC's PSTN telephone switch into a Mobile Switching Center (MSC) for Motorola to sell as a part of their cellular product offering in the U.S. and many other countries.

8. In 1991, I began working as a consultant to Motorola for its "SuperCell" base station product and as a consultant to British Telecom to upgrade its current analog cellular network, and I was one of the network architects responsible for the design and rollout of its Global System for Mobile Communications (GSM) network known as Cellnet. Beginning in the early 1990s, I was responsible for implementation of the SMS service, including working with suppliers of the SMS Center (SMSC), Mobile Switching Centers (MSCs) and cellular phones to define and roll out the functionality that was to be provided.

9. I was personally involved with Nokia, Ericsson, Motorola and other equipment suppliers in this effort.

10. Since 1995, I have also provided second generation (2G) and third generation (3G) Code Division Multiple Access (“CDMA”) network architecture and equipment design and implementation consulting services to companies such as Sprint, Nextel, Nokia, and Ericsson. While consulting to Nextel, which has since become part of Sprint, as one of the network architects for its iDEN network, one of my responsibilities was to define the network and mobile phone functionality required to support MMS and advanced data communications capability.

11. I am a member of the Institute of Electrical and Electronics Engineers (IEEE), including the IEEE Standards Association. I am also a member of the Association for Computing Machinery (ACM). While employed at DSC, I was a member of the ANSI T1 and T1X1 standard groups responsible for the definition and standardization of the Advanced Intelligent Network (AIN) and Signaling System 7 (SS7) protocol.

12. I am currently the president of two consulting companies: Telecom Architects, Inc. and Reticle Consulting, LLC. Telecom Architects provides consulting services to fixed and wireless telecom service providers and their equipment suppliers. I have been President of Telecom Architects since 1999.

13. My experience specifically relevant to this declaration is that I have worked on the design of packet network topologies and their routing algorithms, the

development of computer systems and communication protocols for packet networks since 1976, which is before the current Internet was “invented” in 1983. Specifically, I began this work when I was assigned to a special project associated with the ARPANET while I was serving in the U.S. Army Signal Corp.

14. Since 1976, I have been involved with the design, development and implementation of many of the different types of computer systems and communication protocols used in the Internet. While working at Digital Switch Corporation (DSC), the engineering team that I managed designed and developed the software and hardware from scratch for at least the TCP/IP, X.25, Session and Transport communication protocols so the DSC telephone switches could connect to packet-based networks. Later, while working for Tandem Computers, my development team wrote the software for the HTTP communication protocol so that our computer systems could provide Internet server functionality.

15. Beginning in mid-2007, I worked with Google regarding the design and development of its extensive Content Delivery Network, named the Google Global Cache (GGC), that was mainly used at the time to cache Search requests/responses. During this time, Google had at least 500,000 servers in its network and used a multi-tier topology. Akamai Technologies, Inc., referenced by multiple prior art references, was one of Google’s competitors.

16. As a technical expert on legal cases, I have evaluated various Internet packet routing methods, Internet network topologies, high-speed Internet routers, client/server computer systems, cellular network and Internet interworking protocols, multi-media streaming across the Internet, specialized systems and methods for redirecting Internet users, and specialized CDNs for supplying real-time marketing data and transfer of very large files.

II. MATERIALS CONSIDERED

17. In developing my opinions below relating to the '053 patent, I have considered the '053 patent, its prosecution history, and any materials cited throughout my declaration.

III. LEGAL STANDARDS

18. Counsel has informed me of the legal principles that apply in determining whether patent claims are unpatentable for obviousness. Paragraphs below beginning with “I understand,” “I have been informed,” or similar phrases capture my understanding of the law as provided to me by counsel. The other paragraphs are my opinion.

19. I understand that a claim may be unpatentable as obvious under § 103(a) if the subject matter described by the claim as a whole would have been obvious to a hypothetical “person of ordinary skill in the art” (“POSITA”) in view

of a prior-art reference, or combination of prior-art references at the time the claimed invention was made.

20. I understand that “the time the claimed invention was made” is often referred to as the priority or critical date.

21. I understand that, before considering the prior art, the claims are first construed to determine their scope. I understand that the claims are read from the perspective of POSITA at the time of the alleged invention. I understand that the words of a claim are generally given their ordinary and customary meaning as understood by a POSITA in light of the entire patent specification in which they appear, as well as the prosecution history of the patent.

22. I understand, however, that the words of a claim may be given a meaning other than their plain and ordinary meaning if: (1) a patentee expressly and unambiguously defined a term in a way that is different from its plain meaning; or (2) a patentee “disclaims” the full scope of a claim term’s plain meaning, for instance, by making clear and unambiguous statements during prosecution that a claim term does or does not encompass certain features.

23. I understand that, after construing the claims, the teachings of the prior art are considered. I understand that the prior art includes references that were known or used by others in this country, or patented or described in a printed publication in

this or a foreign country, before the date the invention was allegedly made by the patent holder. For a printed publication to qualify as prior art, I understand that the Petitioner must demonstrate that the publication was disseminated or otherwise sufficiently accessible to the public. However, I also understand that a POSITA is assumed to be aware of all relevant prior art in the field of endeavor covered by the patent-at-issue and all analogous prior art.

24. I understand that although the ultimate question of the obviousness of a claimed invention is a legal determination, obviousness is based on several factual inquiries, including: the scope and content of the prior art, the differences between the claimed subject matter and the prior art, the level of ordinary skill in the art at the time of the invention, and any “objective indicia” or “secondary considerations” of non-obviousness, which must all be considered.

25. I understand that, in determining the scope and content of the prior art, a reference is considered relevant prior art if it falls within the field of the inventor’s endeavor, or if it is reasonably pertinent to the particular problem that the inventor was trying to solve. A reference is reasonably pertinent if it logically would have commended itself to an inventor’s attention in considering his problem.

26. I understand that, to assess the differences between prior art and the claimed subject matter, the claimed invention must be considered as a whole. I

understand that this involves showing that a POSITA, confronted by the same problems as the inventor and with no knowledge of the claimed invention, would have identified the elements in the prior art and combined them in the claimed manner.

27. I understand that there are several rationales for combining prior-art references. For instance, I understand that it is considered obvious to:

- combine prior-art elements according to known methods to yield predictable results;
- substitute one known element for another to obtain predictable results (also called a “simple substitution”);
- use the prior-art elements in a predictable way according to their established functions;
- apply a known technique to a known device (method or product) ready for improvement to yield predictable results (also called a “known technique”);
- choose from a finite number of identified, predictable solutions, with a reasonable expectation of success (also called “obvious to try”);
- or if there is some teaching, suggestion, or motivation in the prior art that would have led a POSITA to modify a prior-art reference or

combine prior-art teachings to arrive at the claimed invention (also called “teaching-suggestion-motivation”).

I understand, however, that a POSITA does not need to have the same motivations or reasons to combine the prior art as the inventors, and can arrive at the claimed invention for completely different reasons than the inventors.

28. However, I also understand that the prior art cannot be modified or combined using “hindsight.” I understand that hindsight refers to situations in which the patent-at-issue is used as a framework to pick and choose pieces from the prior art to arrive at the claims, or adopting rationales identified by the patent-at-issue that those in the art would not have recognized. Instead, I understand that the obviousness inquiry is evaluated from the perspective of a POSITA as of the critical date of the patent-at-issue, with all the prior art before them, to determine whether the POSITA would have independently arrived at the claimed invention without the benefit of the patent-at-issue for guidance.

29. I understand that “objective indicia” or “secondary considerations” of non-obviousness include whether: (1) there was a long-felt need for the claimed invention; (2) the claimed invention has achieved commercial success; (3) there was copying of the claimed invention by others; (4) there were failed attempts by others to make the alleged invention; (5) there was praise of the invention in the field; and

(6) there was skepticism the claimed invention could be achieved, among a few others.

30. As of the time that I prepared this declaration, the Patent Owner had not identified any secondary considerations of non-obviousness. Therefore, I reserve the right to elaborate on my understanding of these secondary considerations and to address any secondary considerations that Patent Owner may subsequently identify.

31. Finally, I understand that, in an inter partes review proceeding, the obviousness of a claim must be demonstrated by “a preponderance of the evidence,” and that this burden falls on the Petitioner. I understand that the preponderance of the evidence standard requires that a reasonable factfinder could find a material fact is more probable than the nonexistence of that fact. It does not allow for speculation regarding specific facts and is instead focused on whether the evidence more likely than not demonstrates the existence or non-existence of specific material facts. I understand that “preponderance of the evidence” is a lower standard than “clear and convincing evidence” (which requires a fact to be substantially more likely to be true than untrue), or “beyond a reasonable doubt” (which is an exceedingly high standard that I understand is generally reserved for criminal matters).

32. I have applied the “preponderance of the evidence” standard throughout my declaration.

IV. TECHNOLOGY BACKGROUND

33. The purported invention of the '053 Patent relates to content delivery and content delivery networks (CDNs). “More specifically, to content delivery networks and systems, frameworks, devices and methods supporting content delivery and content delivery networks.” *See '053 patent* (Ex-1001) at 1:59-62, where it describes the “Field of the Invention.”

34. This section describes, at a high level, what a CDN is, typical CDN frameworks and components, how a CDN fits in the Internet and the protocols used to implement a CDN.

A. The World Wide Web and the Internet

35. At the inception of the World Wide Web (WWW), aka the Internet, in the early 1990s and up to the early 2000s, when an end user (client) wanted to access specific content over the Internet, they would typically use the browser on their computer to retrieve specific web pages, data, stream music or movies, etc. These requests would use a Uniform Resource Locator (URL) that was used to deliver the request to the correct web server.

1. Domain Names and URLs

36. These web servers were typically computer systems that had high-speed communication interfaces and databases for organizing and storing their content. In order to route the client's request to the correct web server over the

Internet that consisted of thousands of potential routers and paths, each of these servers was assigned a unique Internet Domain Name that was included in the URL.

We all have used URLs with different Domain Names when we wanted to access many different types of data on the Internet. For example, <http://www.example.com/home> is a URL that consists of three main parts. The first part is called the “scheme” and it describes the protocol being used to access the server. For this example, the “http://” defines that the HTTP communication protocol is to be used (described in more detail below). The second part is the server’s Internet address, e.g., “www.example.com.” where the “www” stands for world wide web and is a required part of the URL, the [“example.com”](http://www.example.com/) portion is the domain name of the server. The third part of the URL names a specific resource on the web server, “/home” is referencing the home page on the server called “home.” The web server owners or their representative typically maintained the content on their server with updates as they were needed.

2. Web Request and Response

37. The diagram below shows, at a very high level, a client (user with a browser) making a request to a web server for a specific resource and the server responds by providing the resource using the HTTP communication protocol to transfer both the request and response.

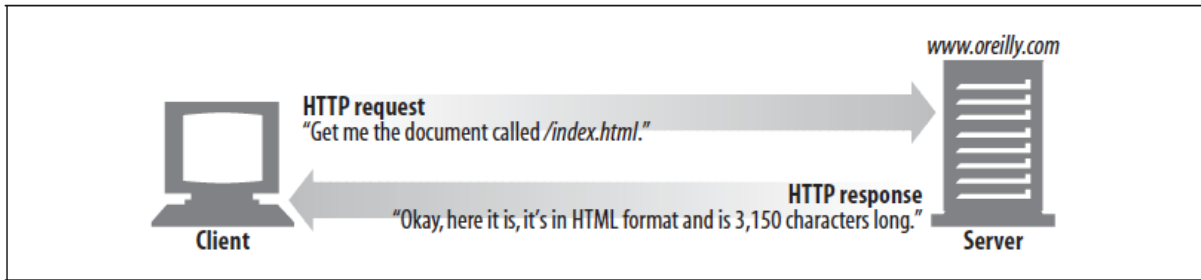


Figure 1-1. Web clients and servers

Gourley, 4.

3. Data Resources/Objects

38. Web servers host web resources (also referred to as objects). A web resource is the source of web content. “The simplest kind of web resource is a static file on the web server’s filesystem. These files can contain anything: they might be text files, HTML files, Microsoft Word files, Adobe Acrobat files, JPEG image files, AVI movie files, or any other format you can think of.” *Gourley*, 4.

4. The HTTP and HTTPS Protocols

39. The Hypertext Transfer Protocol (HTTP) and the HTTP Secure (HTTPS) are request-response protocols that are used by programs to communicate over the Internet. Both are used to provide two-way communication between the web browser (client) and a web server. HTTPS provides essentially the same functionality as the HTTP protocol but it is used to encrypt the plain text messages used by HTTP to prevent eavesdropping or tampering. The diagram below shows this relationship.

a. HTTP/HTTPS Message Formats

40. The structure of HTTP/HTTPS messages include three parts called: 1) the Start line; 2) the Header fields; and 3) the Body. The start line indicates “what to do for a request or what happened for a response.” The Header fields “follow the start line. Each header field consists of a name and a value, separated by a colon (:) for each parsing.” The Body follows the Header field, is optional and can contain “any kind of data. . . . the body can contain arbitrary binary data (e.g., images, videos, audio tracks, software applications). Of course, the body can also contain text.” *Gourley*, 11.

41. The start line of every HTTP message includes a request command called the “HTTP method.” The “method tells the server what action to perform (fetch a web page, run a gateway program, delete a file, etc.).” *Id.*, 8. The table below shows five common HTTP methods.

Table 1-2. Some common HTTP methods

HTTP method	Description
GET	Send named resource from the server to the client.
PUT	Store data from client into a named server resource.
DELETE	Delete the named resource from a server.
POST	Send client data into a server gateway application.
HEAD	Send just the HTTP headers from the response for the named resource.

Id. pp. 8-9.

42. Every HTTP response message is returned with a status code. “The status code is a three-digit numeric code that tells the client if the request succeeded, or if other actions are required. A few common status codes are shown in Table 1-3 below.” *Id.*, 9. The 302 Redirect code is used extensively to route requests to the correct server in a CDN network (described below).

Table 1-3. Some common HTTP status codes

HTTP status code	Description
200	OK. Document returned correctly.
302	Redirect. Go someplace else to get the resource.
404	Not Found. Can't find this resource.

Id., 9.

43. The figure below shows the message details for a typical HTTP transaction between a client that is requesting information about a saw blade from joes hardware that is on special. I note the “Get” method is used in the request and the 200 “Status Code” is used in the response.

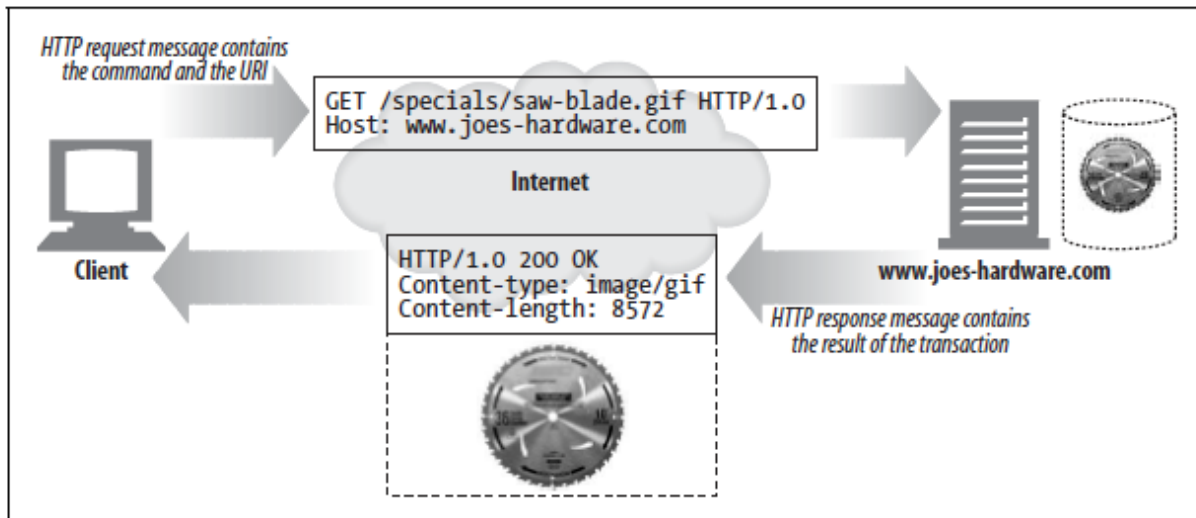


Figure 1-5. HTTP transactions consist of request and response messages

Id., 8.

5. Domain Name Servers and Redirection

44. The Internet routing protocol uses an IP address, which is a series of numbers, to route packets between the client and server and it cannot use the text www.joes.hardware. A special server in the Internet called a Domain Name Server (DNS) is used to convert these textual domain names that are received from the client into the correct IP addresses. A CDN uses a DNS to convert a requested URL into an IP address for an appropriate edge server, e.g., the CDN edge server that is closest to the requesting client. Thus, the request is not delivered to the real web server. Instead, the CDN handles the request, which offloads the real web server (also called the origin server in most CDNs).

45. Redirection is an HTTP method that directs the requesting client to use a different URL for its request by returning the HTTP 302 response code described above along with the new URL it is to use. It can be used for many reasons. For Content Delivery Networks, the Redirect method is typically used to direct the client to a different CDN server for many different reasons that are described in the prior art and would have been understood by a person of ordinary skill in the art.

B. Content Delivery Networks

46. As internet usage grew exponentially in the early 2000s, those in the art recognized that “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.

47. Additionally, significant physical distance between a requesting network element and responding server often leads to greater latency (or delay) between a request and a response. *See generally Buyya*, 45, 51-52. For instance, if a user in Australia tries to stream a movie hosted on a United States-based server, the stream may experience notable delay and buffering as the movie data navigates globe-spanning connections to deliver the movie data to the end user.

48. To solve these problems, companies began building networks specifically intended to distribute content around the world faster while avoiding

overloading a content provider's server. These networks would employ "different levels of local, regional, [and] international caches at geographically distributed locations." *Id.*, 9, 16.

49. CDNs were well-established years before the patent. *See, e.g., 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. Companies including Akamai, Digital Island, and Speedera were first to implement CDNs. *See, e.g., Smith*, Cover (22), ¶5 (describing conventional CDNs in the publication of a patent filed in 2001).

50. The content provider's server was typically referred to as an "origin" server, since it was the originating source of all content for a particular content provider. *Buyya*, 5, 37 ("The server where the definitive version of the content resides is called the origin server. It is updated by the content provider."). The content provider's origin server would then connect to a CDN—typically offered by a third-party company—which hosted "*caching* and/or *replica servers* located in different geographical locations to replicate content." *Id.*, 5. "A cache server makes copies (i.e. caches) of content at the edge of the network in order to bypass the need of accessing origin server to satisfy every content request." *Id.*, 37. These "caching" and/or "replica" servers were also called "edge," "parent," "surrogate," or other

names, depending on their purpose in the network. *See, e.g., id.; see also Outten, Fig. 2.*

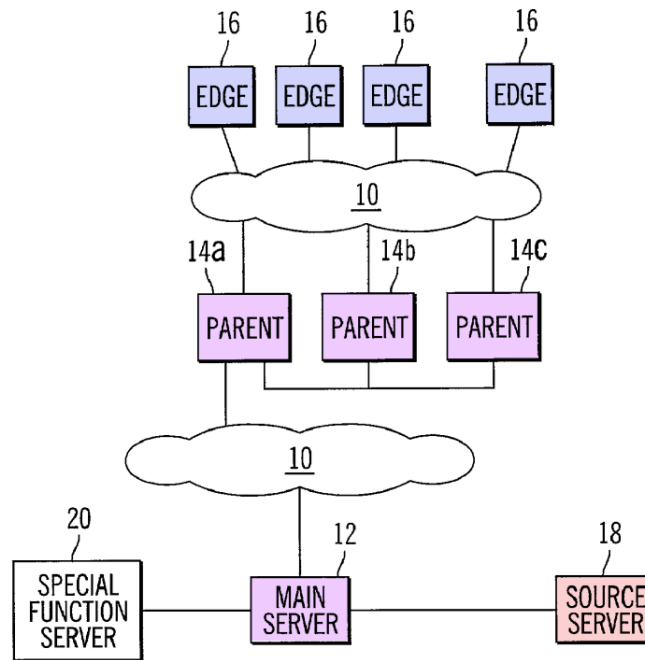


FIG. 2

Outten, Fig. 2 (annotated)

51. To distribute the load of both requests and responses, as well as reduce delivery times, CDNs were often arranged in a hierarchy. *Xu, 57; Buyya, 9.* Fig. 1.1 below is one example of a hierarchical CDN topology. The content provider's origin server (shown in red, below) would sit furthest from the end user in the hierarchy. The CDN would then include, for instance, intermediate servers—called “surrogate” origin servers and/or parent servers (purple)—that may be geographically distributed and would cache at least some parts of the content owner's library. These surrogate-

origin servers and/or parent servers would then support clusters of also-geographically dispersed “edge” or “child” servers located closest to the end user, which would also cache at least part some of the content owner’s library.

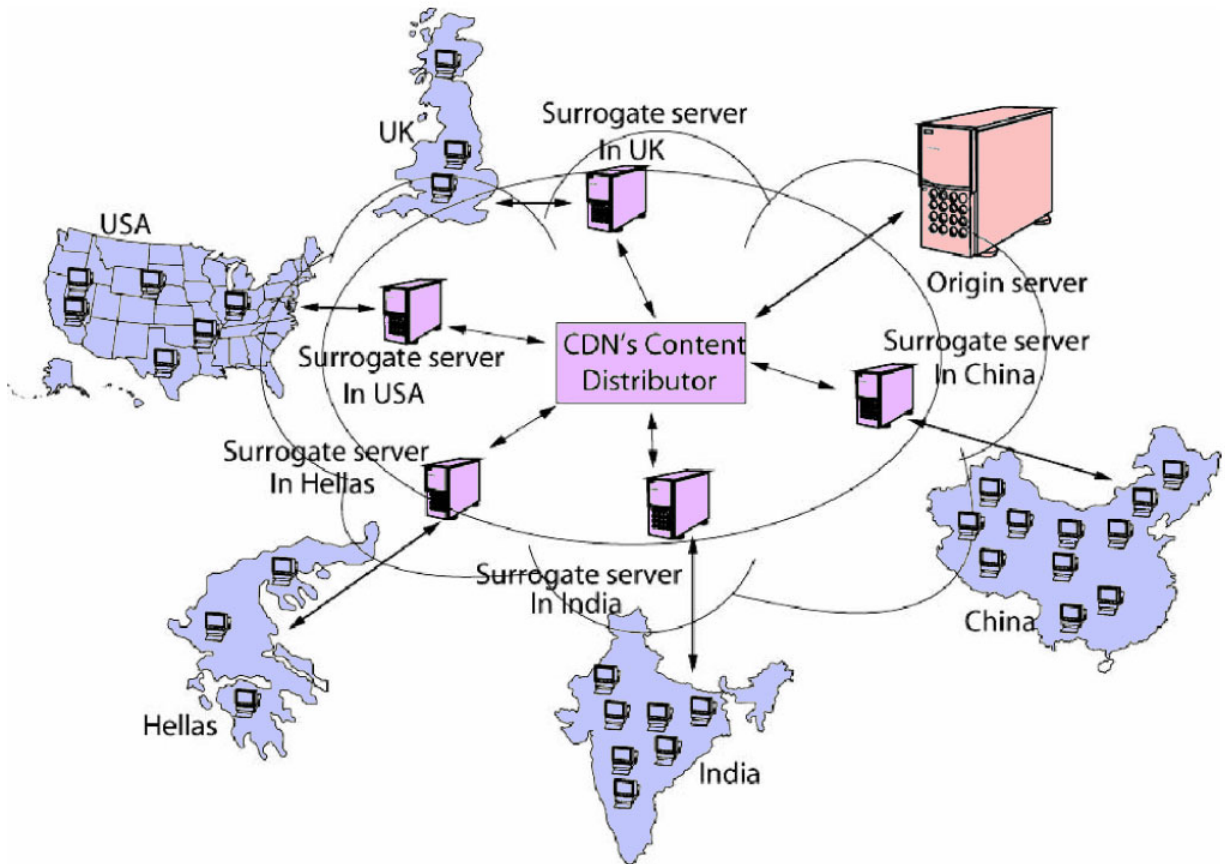


Fig. 1.1 Model of a CDN

Buyya, 5 (annotated).

52. Content would be replicated across servers based on a number of factors. Fig. 2.6 below provides one model for how data could be replicated across servers. For instance, content providers could choose between “full-site” caching (all content is hosted by the CDN) or “partial-site” caching (select content is hosted by

the CDN). *Id.*, 44. When “partial-site” caching is used, what content to host in the CDN can be selected based on empirical (heuristic-based manual selection), popularity (where “the most popular objects are replicated” throughout the CDN), objects (selection of specific types of objects to host), or clusters (hosting content that has been correlated based on specified factors). *Id.*, 44.

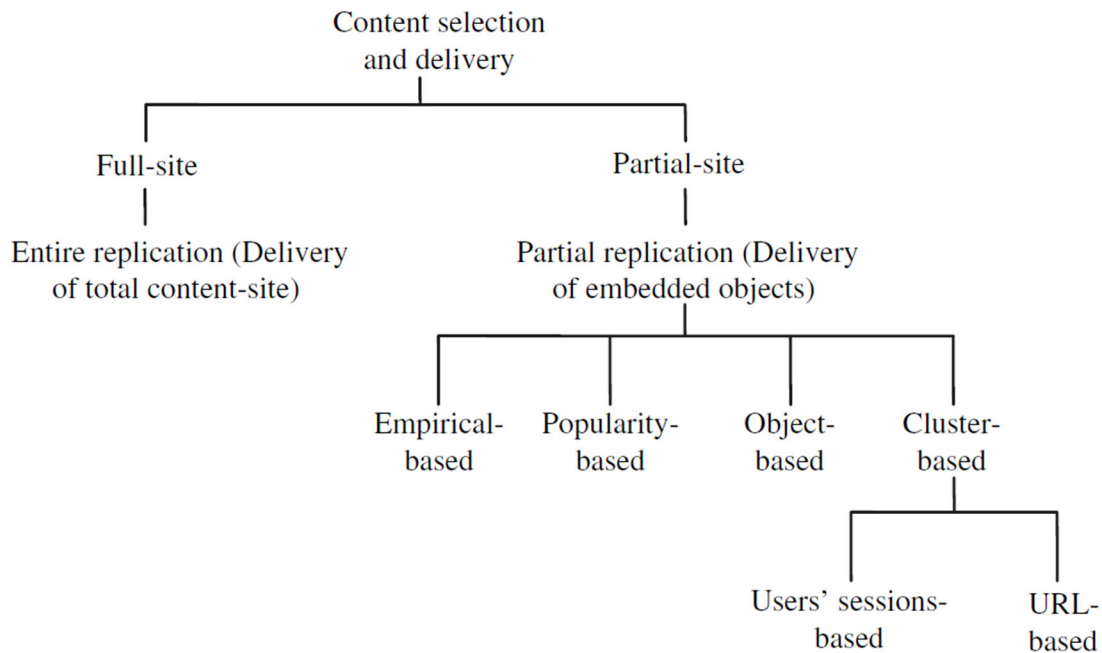
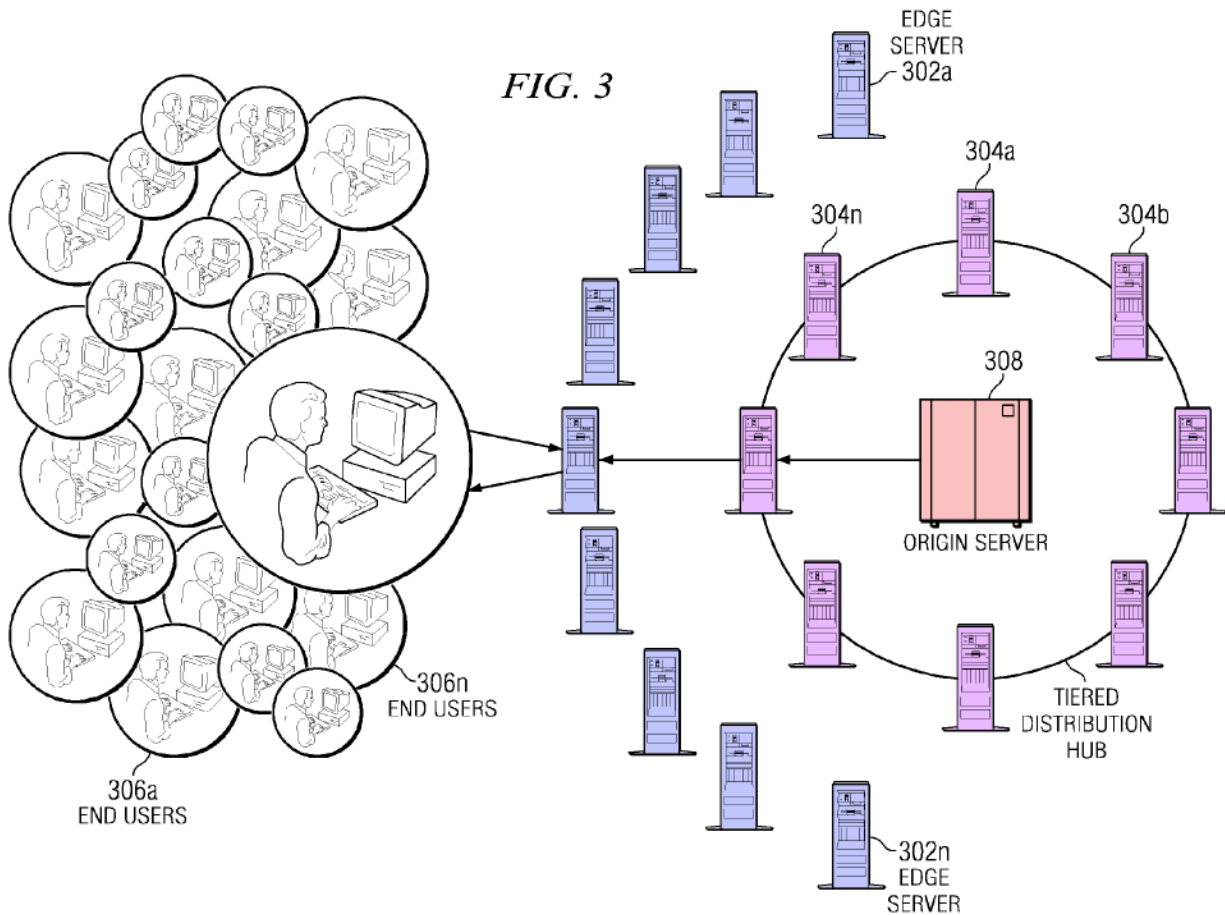


Fig. 2.6 Taxonomy of content selection and delivery

53. Fig. 3 below shows yet a different CDN hierarchical topology that was used. Thus, several different CDN topologies were known, and the exact configuration, including the number of servers, placement of servers, storage procedures, request-routing procedures, and delivery procedures depended “on the content/services it provides to its users,” and factors such as cost, quality of service,

latency, redundancy, and others. *Id.*, 36; *see also id.*, 37-58, 63-64, 67-69 (describing considerations for each choice and providing example choices by certain CDN providers), 58-59 (summarizing key metrics that impacted design choices).



Kovacs, Fig. 3 (annotated),¹ 3:5-9, 4:9-35.

54. In hierarchical systems comprising edge, parent, and origin servers, when an end user requests the delivery of specific content, the request would

¹ I added all annotations unless I note otherwise.

typically be routed to the optimal edge server that would try to fill the request itself, minimizing the delivery time and avoiding burdening the content owner's origin server. *Buyya*, 6; *Dilley*, 50. But if the edge server could not locate the content in its cache, the edge server could either search for the content on, or redirect the request to, another server (such as a parent or surrogate origin server) that may store the requested content instead. *Xu*, 57.

55. “To locate content in CDN peering, a centralized directory model, Distributed Hash Table (DHT), flooded request model, or document routing model can be used.” *Buyya*, 58. A centralized directory model uses a centralized directory to lookup which server should be storing the requested content. *Id.* This approach, however, “is subject to a single point of failure,” and scalability “is limited to the capacity of the directory.” *Id.* With DHTs, “peers are indexed through hashing keys within a distributed system,” which effectively builds in “load balancing by offloading excess loads to less-loaded peers.” *Id.* Flooded request models involve servers that propagate requests to all of their connected peers until one peer in the network responds saying it has the requested content. *Id.* But this approach has drawbacks in that “it generates unnecessary network traffic and hence, it requires enormous bandwidth.” *Id.* Finally, the document routing model uses requests that

contain a file identifier, which is forwarded to a peer whose identifier is most similar to the file identifier, which locates the file and provides it to the requesting peer.” *Id.*

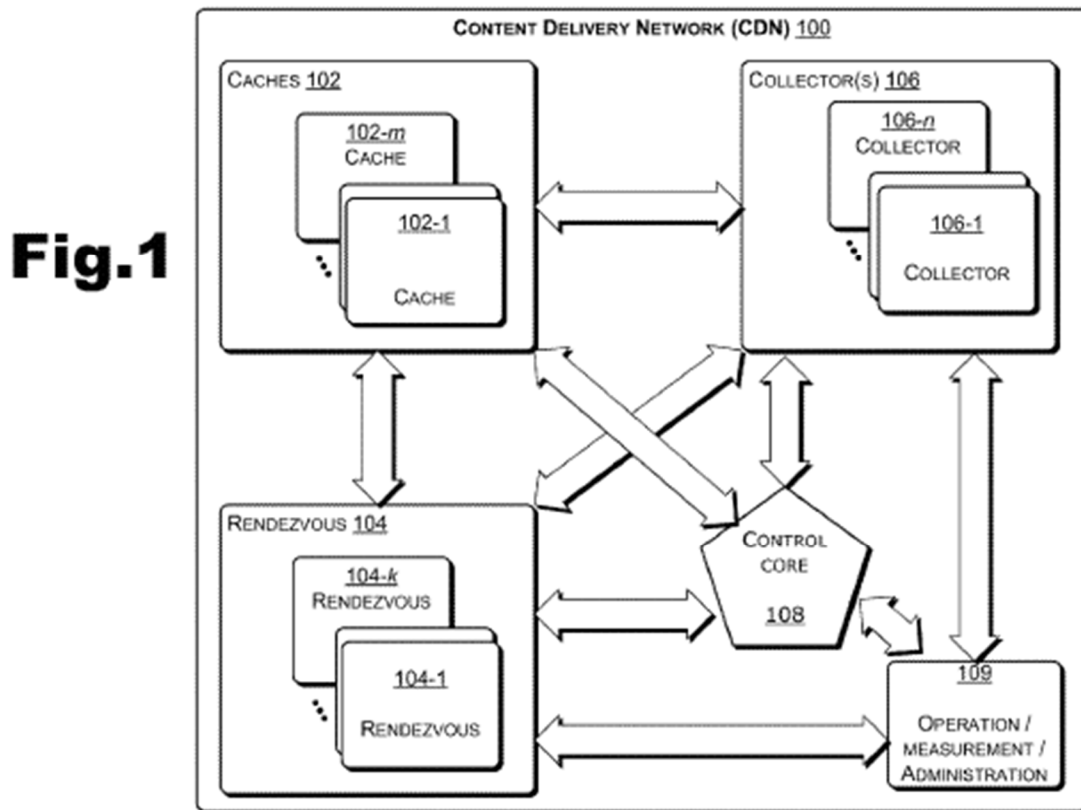
56. If the requested content still could not be located, only then would the request be routed to the content provider’s origin server.

V. ’053 PATENT

A. Specification

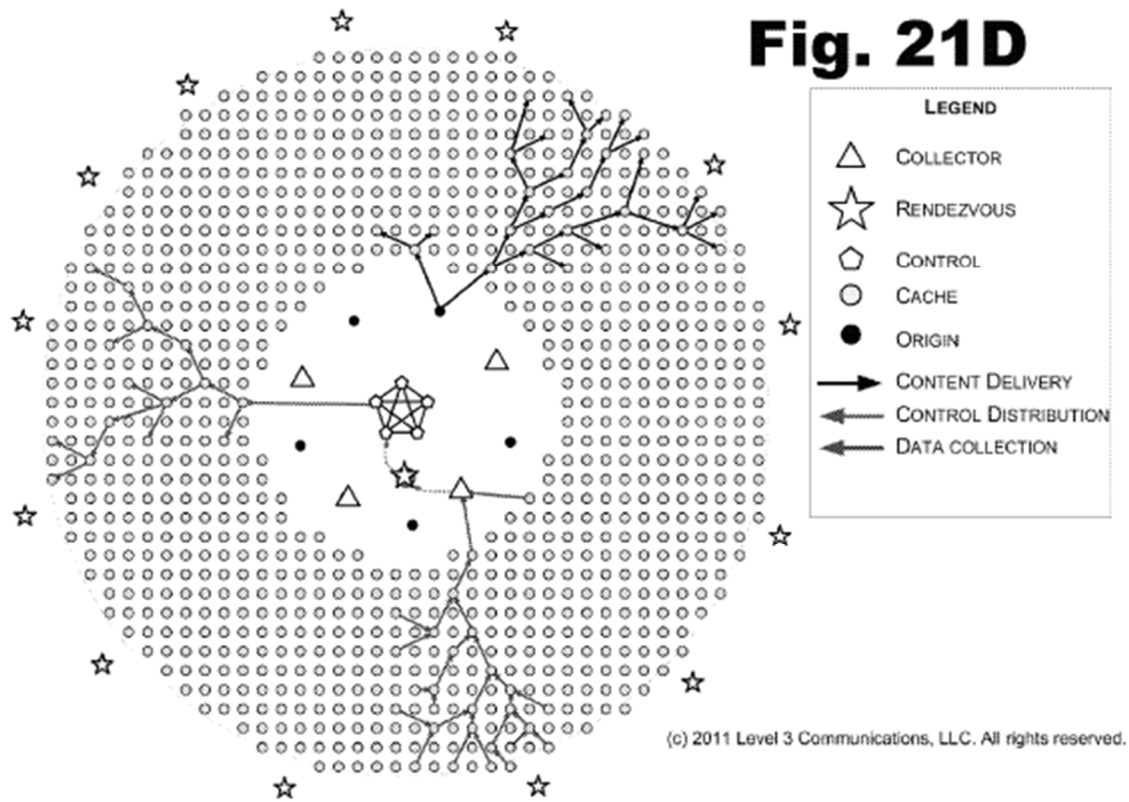
57. I see that the ’053 patent is entitled “Content Delivery Network” and was filed December 14, 2012, with alleged priority to December 14, 2011. Ex-1001, Cover. The ’053 patent discloses configuring CDNs, in which CDN operators control how the CDN is configured, generally, using “global” configurations, 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.

58. The '053 patent discloses other elements, too, 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.



Id., Fig. 1.

59. The '053 patent also discloses optimizing the network using content and control data flows and feedback flows:



Id., Fig. 21D.

60. By example, the '053 patent discloses optimizing CDN performance by employing: (i) a control core distributing control data (containing global configuration and customer configuration data) to cache servers specifying how to fulfill content requests; (ii) a rendezvous mechanism that selects the best cache server to fulfill a particular customer content request, and (iii) feedback logs recorded by network performance collector mechanisms that report real-time network operations and enable subsequent routing decision updates. *Id.*, 3:24-32, 4:43-50; 40:47-21, 42:10-27.

B. Prosecution History

61. I have reviewed the prosecution history of the '053 patent at the U.S. Patent office in forming my opinions.

62. The '053 patent was filed as U.S. Application No. 13/714,410 (“’410 application”) on December 14, 2012, and alleges priority to U.S. Provisional Application Nos. 61/570,448 and 61/570,486 filed December 14, 2011. Ex-1001, Cover.

63. During prosecution, I see that the Examiner rejected all claims of the '053 Patent once for double patenting over then-U.S. Application No. 13/714,411, anticipation (U.S. 7,010,578 to Lewin), and obviousness (Lewin in view of Official Notice). Ex-1002, 523-40. Responsively, the then-patent-applicant (or Patent Owner, which I abbreviate as “PO”) filed a terminal disclaimer, amended the claims, and argued that Lewin delivers cached content directly to a *customer* via a delivery address distinguishable from the amended claim language (below) that delivers cached content to a *client of* a customer more broadly defined as a subscriber to the CDN or another CDN component:

1. (Currently Amended) A content delivery network (CDN), the CDN serving resources on behalf of one or more customer of the CDN, the CDN comprising...

...(b)(2)(3) serve to a client a particular resource associated with a particular customer of the CDN...

Id., 592. *See also id.*, 592-601 (amended claims), 602-06 (PO arguments), 612 (terminal disclaimer).

64. The Examiner then allowed the claims without any substantive explanation. *Id.*, 630-35. However, as I explain below, it is my opinion that Lewin did in fact teach “customers” and “clients” consistent with the claims, and that the Examiner failed to combine Lewin with other readily available prior art, which would have supported a continued rejection.

VI. LEVEL OF ORDINARY SKILL

65. I understand that when both interpreting the claims and assessing the prior art, I must do so from the perspective of a person of ordinary skill in the art (“POSITA”) as of the relevant priority date. My understanding is that the earliest possible priority date of the ’053 patent is December 14, 2011.

66. In my opinion, in the 2011 timeframe, a POSITA 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. However, work experience can substitute for formal education and vice versa.

67. I was at least a POSITA as of the earliest possible priority date of the '053 patent, and still am.

VII. CLAIM CONSTRUCTION

68. I have applied the plain and ordinary meaning to all terms of the '053 patent as would have been understood by a POSITA on or before December 14, 2011.

VIII. SUMMARY OF OPINIONS

69. In my opinion, the claims of the '053 patent are obvious in view of the following combinations of prior art:

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

70. In my opinion, all the prior art is analogous to the '053 patent because it is all within the same field of endeavor and reasonably pertinent to the problems

face by the inventor of the '053 patent (e.g., configuring CDN networks for content delivery). *Compare, e.g., '053 patent*, Abstract (“[a] content delivery network”) *with Lewin*, 2:44-55 (“the content delivery service (CDS) comprises a preferably global content delivery network (CDN)”; *Wein*, Abstract (“serv[ing] content through a content delivery network (CDN)”; *Liu*, ¶ 28 (“The UDN has a combined content delivery network 103”); *Devanneaux*, ¶11 (“CDN [content delivery network] edge server is configured to provide one or more extended content delivery features on a domain specific, customer-specific basis”); *Nygren*, 2 (“Akamai first pioneered the concept of Content Delivery Networks (CDNs)”; *Lewin819*, Abstract (“The disclosed technique enables a content provider to dynamically assemble content at the edge of the Internet, preferably on content delivery network (CDN) edge servers”); and *Crosbie*, 5:1-9 (“The present invention relates to global Internet performance and the ability to provide users with consistently fast and reliable access to a content provider's data...[with] a global system [that] combines local content caching and mirroring with a traffic management mechanism”).

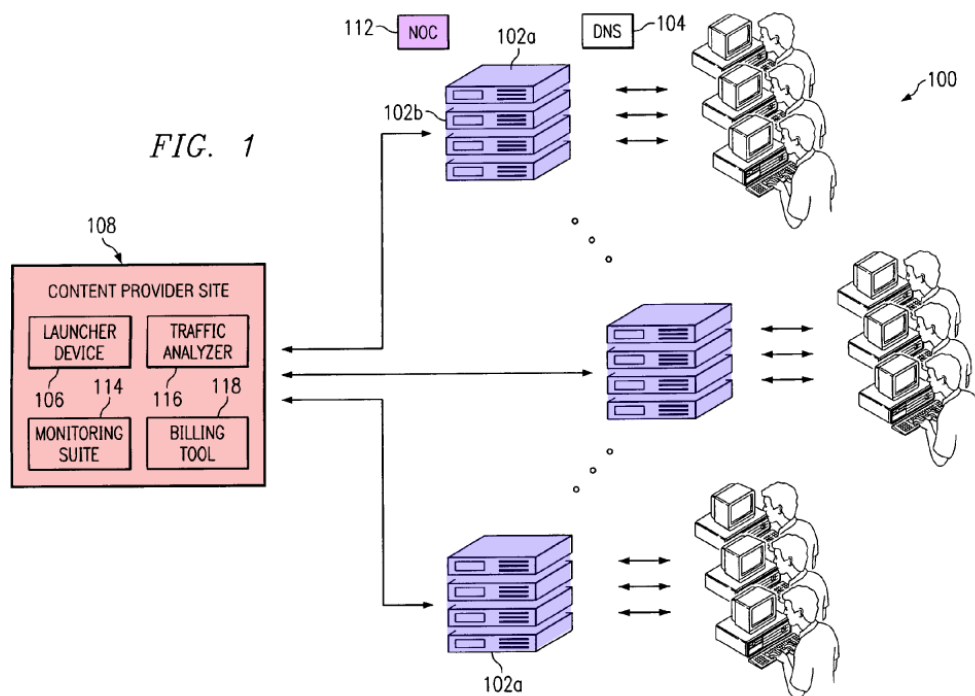
IX. DETAILED EXPLANATION OF GROUNDS

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

71. At a high level, Lewin discloses “a content delivery service to extend the reach of its network.” *Lewin*, Abstract. Lewin explains that the content-delivery service (“CDS”) “comprises a preferably global content delivery network (CDN) 100 of content delivery servers 102a-n.” *Id.*, 2:47-50. Content providers (which I show in 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.



Id., Fig. 1. Some content providers used Lewin’s CDN topology or a similar topology—a well-known CDN topology configured by Akamai Technologies is also the same or very similar—to host their websites. *Id.*, Cover, (73) (assigned to Akamai Technologies), 1:14-25, 3:14-23, 6:9-17.

72. Lewin’s CDN design allows third parties other than the CDN operator to add their own caches (or servers) to participate in the CDN. *Id.*, 1:61-2:12. 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).” *Id.*, 3:43-6:8.

73. For instance, 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.” *Id.*, 4:57-6:3.

74. Also, during registration, Lewin’s third-party cache provides the CDN configuration information, such as whether the cache “may have a ‘public’ status” that “can serve general purpose traffic,” or a “‘private’ status” that permits access only by those from within the cache owner’s enterprise. Moreover, “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.

75. Lewin further 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.” *Id.*, 3:19-23.

b. Wein

76. Wein, like Lewin, discloses a CDN network. *Wein*, Abstract. Wein, however, details various configuration parameters involved in setting up and maintaining the CDN network itself, noting configurations are provided in “metadata.” *Wein*, Abstract, 1:17-52. Specifically, Wein explains how parameters that determine how the CDN content server handles object requests are written into metadata, which is then sent to the servers responding to requests. *Id.*, 2:13-27, 4:62-5:26. This metadata may either be provided within a request itself, or in a separate configuration file. *Id.*, 3:8-23.

2. Motivation to Combine Lewin with Wein

77. Lewin’s focus is on the high-level configurations involved in adding a third-party cache to an existing CDN network, but Lewin is less concerned with the

specifics of 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.

78. In my opinion, a POSITA would have been motivated to combine Lewin with Wein because Wein provides greater detail as to how to configure the primary CDN, apart from configuring third-party caches added to the network that was Lewin's focus. For instance, Wein explains how the primary CDN uses more-global configurations provided by CDN operators as well as separate, customer-specific configurations, and details how ARLs (which Lewin also uses in its CDN) can also include customer-configuration information. *Wein*, 4:39-52, 4:62-5:15, 8:12-31. In other words, the two references are directed to different types of configurations that are operable together; Lewin is primarily concerned with configuring a third-party cache that is **added** to a CDN, and Wein is primarily concerned with configuring the CDN itself.

79. In my opinion, a POSITA would have had a reasonable expectation of success in combining Lewin and Wein because both teach nearly identical CDN system structures, such as the central commands, edge servers, and customer origin servers. *Lewin*, 1:61-2:12, 3:24-58; *Wein*, 2:13-27. Lewin and Wein are both also publications from Akamai Technologies, (*Lewin*, Cover (assigned to Akamai

Technologies, Inc.); *Wein*, Cover (same)), which was one of the well-known CDN providers at the time of the '053 patent, indicating to those in the art that they were likely implementable and compatible together. And as I mentioned previously, a POSITA would have understood that Lewin's and Wein's respective teachings complemented each other—as opposed to requiring some modification of one feature expressly taught in another—because they are directed to configuring different aspects of a CDN network. In other words, modifying Lewin would have involved nothing more than configuring 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.

3. Claims 1, 27

a. 1(pre): A content delivery network (CDN), the CDN serving resources on behalf of one or more customers of the CDN, the CDN comprising:

27(pre): A computer-implemented method, operable in a content delivery network (CDN) comprising:

80. If the preambles are limiting, it is my opinion that Lewin discloses or suggests them.

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

82. In my opinion, a POSITA would have understood that Lewin discloses that “content providers” (*customers*) may utilize a CDN to serve content (*resources*) on their behalf. *See, e.g., id.*, 3:14-23, 6:9-17 (explaining content providers host a website that may point to content hosted by a CDN). In fact, CDNs conventionally serve content on behalf of content providers. *See, e.g., Liu*, ¶29 (“Customers can leverage the size, scope, and location of the UDN to store content...for fast and highly available access by clients.”), *Wein*, 1:18-22 (“A CDN is a self-organizing network...arranged for efficient delivery of digital content...on behalf of third party content providers.”).

b. 1(a): a control core; and

27(pre)(a): a control core; and

83. In my opinion, Lewin discloses or suggests 1(a) and 27(pre)(a).

² I italicized text claim text unless otherwise noted.

84. Lewin discloses the CDN includes a “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.

85. In my opinion, a POSITA would have understood that Lewin’s NOC and registration server collectively constitute a *control core* as recited in the claims because they generally control the operation and access to the CDN. *Id.* This is consistent with how the ’053 patent describes the *control core*. ’053 *patent*, 4:43-44 (“The control core mechanism 108 controls operation of the CDN”).

c. 1(b): a plurality of cache servers, each particular cache server of said plurality of cache servers being constructed and adapted to:

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:

86. In my opinion, Lewin discloses or suggests 1(b) and 27(pre)(b).

87. 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. And for claim 27(pre)(b) in particular—which also recites *the CDN serving resources on*

behalf of customers—I previously described how Lewin teaches this element in the context of claim 1(pre) that has the same requirement. See §IX.A.3.a (1(pre)/27(pre)).

88. Lewin also discloses that “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. In my opinion, based on these teachings, a POSITA would have therefore understood that Lewin’s third-party caches comprise computer hardware (*constructed/hardware*) and programmed to serve content and interact with the CDN (*adapted/software*).

**d. 1(b)(1): upon joining the CDN,
1(b)(1)(1): obtain global configuration data from the
control core; and**

**27(A): obtaining global configuration data from the
control core;**

89. In my opinion, Lewin, alone or combined with Wein, teaches or suggests 1(b)(1)-1(b)(1) and 27(A).

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

91. 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*). *Id.*, 4:57-6:3.

92. This is consistent with Wein, which 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.

93. In my opinion, 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. *See* ’053 patent, 14:63-15:19. In other words, these configurations are customer- and object-agnostic, and are intended more to govern operation of the CDN (e.g., globally).

94. Although Lewin does not expressly state which structure provides the configuration information to the third-party cache, in my opinion, a POSITA would have understood that either the NOC or registration server (*control core*) provide the configuration information, because these are the elements Lewin describes as controlling the operation of and access to the CDN. This is again consistent with

Wein, which expressly teaches using a “central authority” to push CDN-generated metadata configuration files. *Wein*, 8:3-5.

e. **1(b)(2): having joined the CDN,
1(b)(2)(1): obtain updated global configuration data,
if needed; and**

**27(B): selectively obtaining updated global
configuration data; and**

95. In my opinion, Lewin, alone or combined with Wein, teaches or suggests 1(b)(2)-1(b)(2)(1) and 27(B).

96. Lewin discloses that, during registration, 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). And like Lewin, Wein also 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, 8:27-29.

97. In my opinion, a POSITA would have understood based on Lewin, alone or combined with Wein, that after joining the CDN, third-party caches receive updated global-configuration data (including, for instance, 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. In fact, it would make little sense to not provide CDNs a way to update their global configuration files, such as to add new servers, or define new operational parameters for the existing network servers.

98. However, if “selectively obtaining” (cl. 27) requires that the cache server *pull* updated global configuration files, in my opinion, that also would have been obvious to a POSITA. Pulling and pushing are two predominant ways of acquiring data across a network; pushing involves one network element sending information to a receiving network element, and pulling involves a receiving network element requesting information be sent from a sending network element. *See, e.g., Reynolds*, ¶¶132-34 (describing how push, pull, and push-pull schemes were the “three basic kinds of algorithms” to distribute information in a network); *Winter*, ¶103 (“the downloading of the server configuration is carried out in Push or

Pull.”). Wein, for instance, 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. In my opinion, because Lewin does not expressly disclose how updated configuration information is communicated between the registration server and third-party cache, a POSITA would have understood that Lewin’s system would obviously be configurable to do either a push or pull or both. In fact, a POSITA would have also been specifically motivated to configure Lewin’s cache servers to do both to add flexibility to the network. For instance, updated configuration information could first be pushed to sets of intermediate servers, and then individual third-party cache servers could pull the updates from those intermediate servers at their convenience, such as during low-traffic points. And given the prevalence of both pushing and pulling techniques, implementing either or both techniques would have involved minimal coding that, in my opinion, would have been well within the skills of a POSITA to implement.

f. 1(b)(2)(2): obtain customer configuration information associated with at least one customer of the CDN;

27(C): obtaining customer configuration information associated with a particular customer of the CDN;

99. In my opinion, Lewin alone or combined with Wein discloses or suggests 1(b)(2)(2) and 27(C).

100. Specifically, Lewin alone or combined with Wein, teaches several different configuration sets that, in my opinion, would have been considered *customer-configuration information* consistent with the claims.

101. 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.

102. Although Lewin discloses this information is initially exchanged during registration, as I mentioned earlier, configuration files are routinely updated, and thus the third-party cache may obtain updated configuration information after joining the CDN. For example, in addition to ordinary updating to account for changed configurations, which I describe above (§IX.A.3.e (1(b)(2)/27(B)), a POSITA would have understood that the cache may update the configuration files if it ever re-registers itself with the CDN because the validity of the registration expires. *Id.*, 5:31-36 (“the cache may need to re-register to keep itself current” once a TTL, or time to live, metric expires).

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

104. Based on these teachings, in my opinion, 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.

105. ***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. *Id.*, 3:53-56, 4:8-26. Although Lewin envisions that *anyone* can theoretically add a third-party cache to the network, in practice, it would have been expected that only those participating in the network would add third-party caches, such as customers or internet-service providers (“ISPs”) to either. For instance, ISPs may add servers at their own server sites to offload requests from passing through their general-purpose networks and direct them instead to the CDN. Additionally, customers may add third-party caches at their corporate headquarters to help assess delivery of their content through the CDN or provide greater access to their content through the CDN at headquarters, or to supplement the CDN in geographies where the customer has clients with high demand for content, but the CDN does not provide

substantial existing server support. When customers provide the third-party caches, a POSITA would have understood that the customer would provide the configuration file associated with that third-party cache.

106. This said, in my opinion, a POSITA would have also understood that CDNs often incorporate additional customer-provided configuration files that allow customers to configure how a CDN will handle the customer's content within the CDN itself. This is different than configuring how a third-party cache will operate within an existing CDN network, like Lewin describes. 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.

107. In my opinion, a POSITA would have understood based on the teachings of Lewin and Wein that their CDNs provide several different ways for

customers to configure both the structure of the CDN and their content hosted by the CDN. Lewin teaches configuring third-party caches to interact with the CDN, but a POSITA would have understood that Lewin would have, or would have 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.

108. In my opinion, 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. As I mentioned, Lewin's customer-configuration information is directed to specifically configuring third-party caches, whereas Wein's customer-configuration information and files are directed to controlling the delivery of customer content over the CDN. Because these two sets of configuration information are directed to different purposes, in my opinion, a POSITA would have appreciated that they would both be implemented in a combined system, and would be compatible with one another rather than conflicting with one another.

109. **ARLs.** Lewin also mentions that the CDN utilizes ARLs to direct clients to the CDN. *Lewin*, 3:19-23. Lewin does not dive into the details as to what an ARL is, but a POSITA would have known that ARLs had a particular meaning for at least the well-known Akamai CDN topologies as referring to a specific type of "Uniform

Resource Locator” (URL) that was commonly used in internet networks that are specifically adapted for CDNs, and also routinely had configuration information incorporated within them.

110. 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.

111. In my opinion, a POSITA would have therefore understood that Lewin’s ARLs would have, or would have 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. §IX.A.2 (motivation).

- g. 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.**

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.

112. In my opinion, Lewin discloses or suggests 1(b)(2)(3) and 27(D).

113. Lewin discloses a process of serving a customer's content to a client. Specifically, it involves making a request of the customer's website, and then having the website direct the client (through the ARLs) to the CDN, including third-party caches. *Lewin*, 3:7-42.

114. Because Lewin, alone or combined with Wein, teaches that the CDN and third-party caches have both been configured using *global-configuration data* (e.g., that specifies both URL maps and parsing information to locate the requested content) and *customer-configuration information* (e.g., that specifies 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), §§IX.A.3.d-IX.A.3.f (1(b)(1)-1(b)(2)(2) and 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).

4. **Claim 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 28: The method of claim 27 further comprising:
(A)(2) obtaining data from one or more other locations in the CDN.**

115. In my opinion, Lewin discloses or suggests claims 2 and 28's additional limitations.

116. Lewin discloses that, upon a third-party cache joining the CDN (cls. 2 and 28),³ the CDN is then used as “an authoritative surrogate content server, just like any other CDN content server.” *Lewin*, 3:49-53.

³ I note that claim 2 specifies certain functions “upon [a cache server] joining the CDN,” but claim 28 is not limited to “upon joining the CDN.” Therefore, I first address functionality “upon joining the CDN,” which applies to both claims 2 and 28, and then functionality that also meets the claim *after* joining the CDN, which applies only to claim 28.

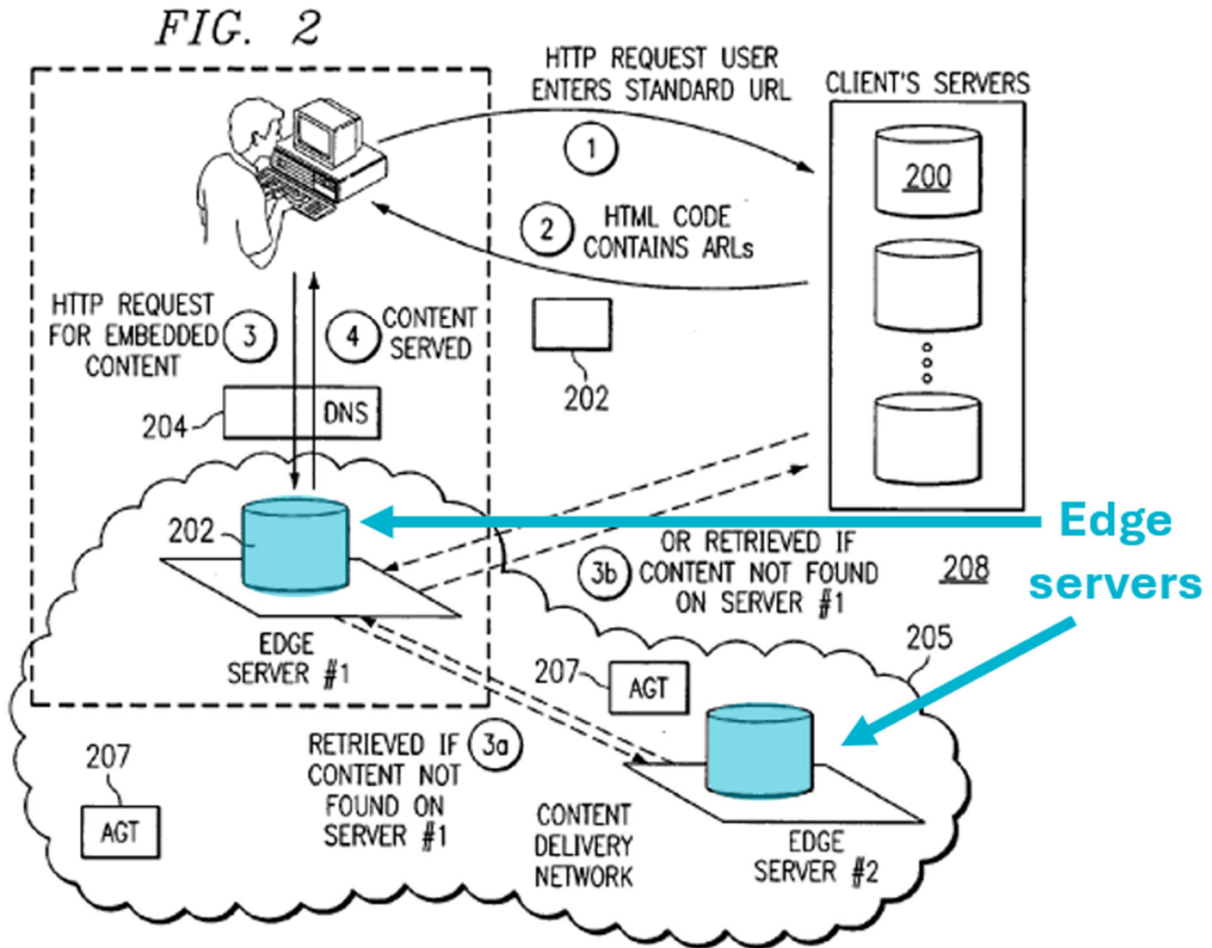
117. In my opinion, a POSITA would have understood that to act like “an authoritative surrogate content server,” the third-party cache would need to cache content to serve. There are two ways this could happen: (1) either the third-party cache remains empty until it receives requests, and then fetches the content from the CDN in response to requests and then caches the content requested for itself (e.g., fetch-and-fill) or (2) the cache is pre-populated with content before receiving any requests. Pre-population may occur, for instance, using load-balancing servers, a central authority server, or even other peer edge servers that identify specific content for the third-party cache to store. Other prior art, for instance, would define rules in which edge servers (like Lewin’s third-party cache) would be pre-populated with popular content that would be expected to be in high demand. *Swildens*, ¶43 (“An edge server maintains a dynamic number of popular files in its memory for the customer.... [W]hen a file has been requested from an edge server a sufficient number of times to become more popular than the lowest popular stored file, the file is obtained”), ¶130 (“popularity is based on the aggregate data from all of the edge servers that are accessing [a] common storage”).

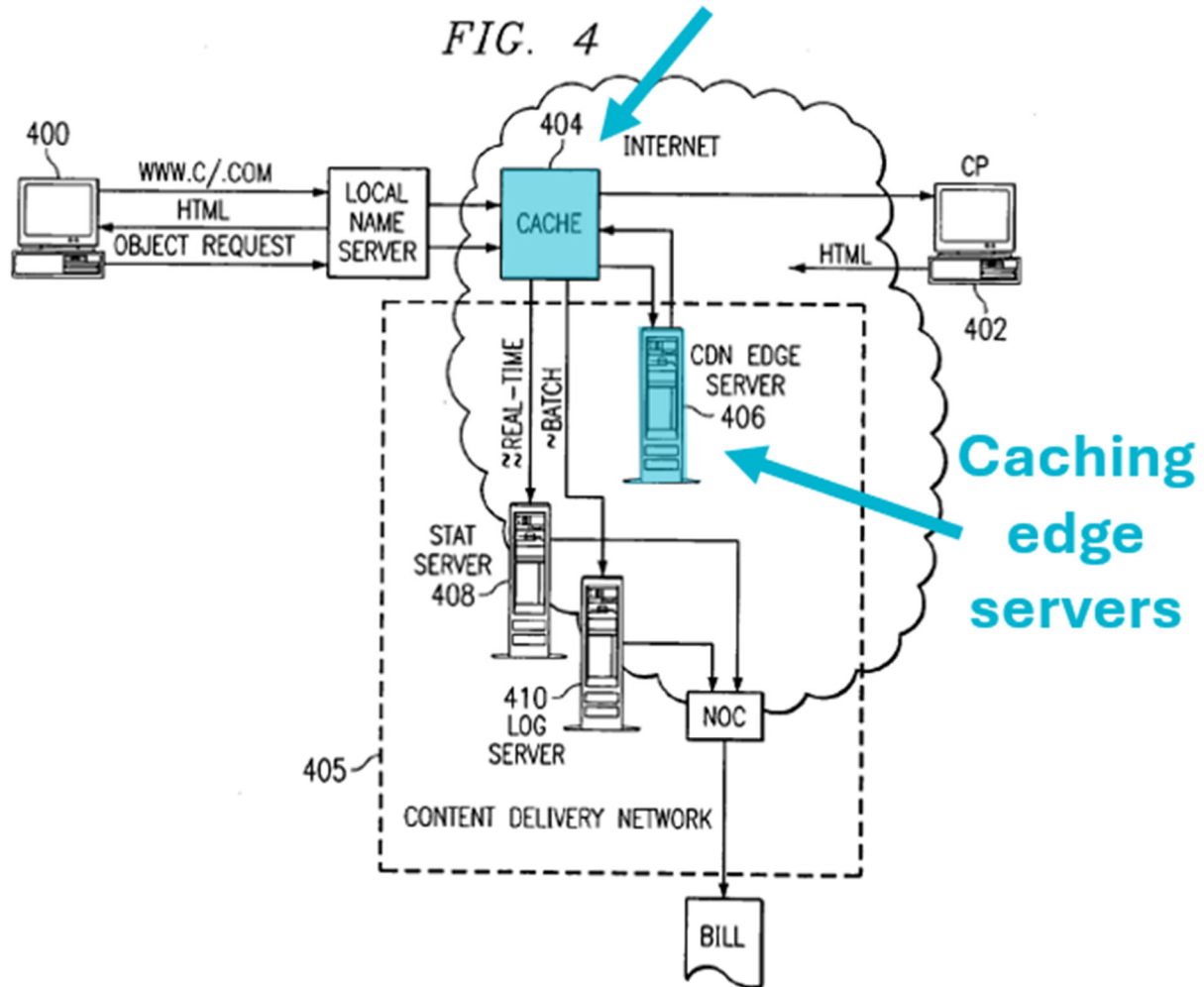
118. A POSITA would have found it obvious, therefore, for Lewin’s third-party caches to pre-populate its cache and obtain copies of content upon joining the CDN (*obtain data from one or more other locations in the CDN*). Moreover, a

POSITA would have expected that the content would be obtained from other CDN caches (e.g., edge caches) already storing the content because there would be expected to be some edge servers closer to the third-party cache than the customer's origin server that could assist in pre-populating the third-party cache, and would avoid overloading the origin server with requests to help pre-populate the third-party cache. *See, e.g., Seed*, ¶62 (“If a popular object has already been replicated to an edge server[,] filling a new edge server request for that object from one of the peer edge servers may be more efficient than the parent server or origin server. Since there are typically more edge servers...there is an increased likelihood that a peer edge server may be closer in terms of network distance... [and] could also lessen the burden on the parent servers or origin servers.”).

119. Consistent with this view, 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. Lewin also depicts this process in Figure 2 (at step 3a), and the arrangement of servers in Figure 4:





Id., Figs. 2, 4.

5. **Claim 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 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.

121. In my opinion, Lewin combined with Wein teaches or suggests claims 3 and 29.

122. As I explain above, Lewin combined with Wein teaches incorporating customer-configuration information associated with a specific customer in ARLs. §IX.A.3.f (1(b)(2)(2) and 27(C)). Lewin also explains that content is served when the cache receives an ARL. *Lewin*, 3:17-23, 6:42-7:28, Fig. 2.

123. Based on these teachings, in my opinion, a POSITA would have 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*.

6. **Claim 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 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.

124. In my opinion, Lewin combined with Wein teaches or suggests claims 4 and 30.

125. As I explain above, Lewin combined with Wein teaches incorporating customer-configuration information associated with a specific customer in a configuration file. §IX.A.3.f (1(b)(2)(2) and 27(C)). And 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. §IX.A.3.f (1(b)(2)(2) and 27(C)). As Lewin explains, these configuration files are exchanged upon adding a third-party cache to the CDN (and may be updated thereafter), (as I explained at §§IX.B.3.e-IX.A.3.f (1(b)(2)-1(b)(2)(2) and 27(B)-27(C))), and thus a POSITA would have understood that the configuration files are processed *prior to receive of a request for a resource associated with the specific customer*.

7. Claim 6: The CDN of claim 1 wherein the customer configuration information comprises at least one CDN resource.

Claim 32: The method of claim 27 wherein the customer configuration information comprises at least one CDN resource

126. In my opinion, Lewin alone or combined with Wein teaches claims 6 and 32's additional limitations.

127. As I explained, 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. §IX.A.3.f (1(b)(2)(2) and 27(C)). In my opinion, a POSITA would have therefore understood that each of these constitute *CDN resources* as claimed because they are resources used by the CDN to deliver content. This is consistent with the '053 patent, which distinguishes between customer resources (e.g., content) and CDN resources. *See '053 patent*, 15:66-16:2.

128. 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.

See id., 40:39-40 (noting a customer-configuration script may “provide...for CDN resources.”). For instance, customer-configuration information may specify to prioritize certain content over other content, or may request specific logging requirements associated with specific content, each of which would either implicitly or explicitly identify specific CDN resources (like priority or logging protocols) to implement.

8. **Claim 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 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.

129. In my opinion, Lewin alone or combined with Wein teaches or suggests claims 8 and 34’s additional limitations.

130. Lewin and Wein both teach that customer-configuration information may specify things like traffic restrictions and registration and validation requirements. *See, e.g., Lewin*, 4:27-56; *Wein*, 4:62-5:10. 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. *See also* §IX.A.7 (cls. 6 and 32) (explaining how customer-configuration

information may explicitly or implicitly define processing requirements). For this reason, in my opinion, a POSITA would have understood the customer-configuration information *specifies customer-specific processing requirements for resources served on behalf of that customer*.

9. Claim 9: The CDN of claim 1 wherein the global configuration data comprises at least one CDN resource.

Claim 35: The method of claim 27 wherein the global configuration data comprises at least one CDN resource.

131. In my opinion, Lewin alone or combined with Wein teaches claims 9 and 35's additional limitations.

132. As I explained, Lewin alone or combined with Wein teaches providing global-configuration information. §IX.A.3.d (1(b)(1) and 27(A)). In my opinion, a POSITA would have understood that the global-configuration file constitutes a *CDN resource* as claimed because they are resources used by the CDN to configure the CDN network. *See '053 patent*, 15:66-16:2 (distinguishing between customer resources (e.g., content) and CDN resources).

133. 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 process and fulfill requests

using CDN-processing resources. *See id.*, 40:39-40 (noting a customer-configuration script may “provide...for CDN resources.”). For instance, Lewin notes that part of the global configuration includes “log recipient and frequency,” real-time “statistics recipient and frequency data,” and “a map URL,” among other pieces of information. *Lewin*, 4:63-5:4. A POSITA would have understood that logging/statistics invoke the CDN’s logging and statistics resources (e.g., running specific logging and statistics protocols that the CDN is evaluating), and the map URL provides an identity of content locations, which would also be considered CDN resources.

10. Claim 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 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.

134. In my opinion, Lewin discloses or suggests claims 11 and 37’s additional limitations.

135. Lewin discloses that the primary CDN includes edge caches (*cache servers*) in addition to those provided by third parties. *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.

136. As I explained, in my opinion, a POSITA would have understood that, upon joining a CDN (cls. 2 and 28) and/or after joining the CDN (cls. 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. §IX.A.4 (cl. 2).

11. Claim 12: The CDN of claim 11 wherein the one or more other locations in the CDN comprise at least one peer location.

Claim 38: The method of claim 37 wherein the one or more other locations in the CDN comprise at least one peer location.

137. In my opinion, Lewin discloses or suggests claims 12 and 38's additional limitations.

138. 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. A POSITA would have considered it obvious that, upon joining a CDN (cls. 2 and 28) and/or after joining the CDN (cl. 28), both third-party cache and edge cache servers would obtain content from their peer caches in the CDN. §IX.A.4 (cls. 2 and 28); *see also, e.g., Seed*, ¶¶20 (noting CDNs typically organized in tiers of origin, parent, and edge servers, each of which may be a “cluster” of servers), 62 (noting peer relationships between different tiers). Moreover, a

POSITA would have understood that because peers are typically more numerous at the lower tier 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. *See, e.g., Seed*, ¶62. In other words, a POSITA would have understood caching content on peer servers offloads that content from the origin server, which “provide[s] better quality of service (QoS) to a broad spectrum of end user systems” around the globe. *Harvell*, Fig. 1, ¶16.

12. Claim 13: The CDN of claim 2 wherein the particular cache server determines said one or more other locations using said information⁴ in said global configuration data.

Claim 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.

139. In my opinion, Lewin discloses or suggests claims 13 and 39’s additional limitations.

⁴ I note that “said information” in Claims 13 and 39 was not clearly introduced in an earlier claim, so I have treated it as global configuration *data* for purposes of this declaration consistent with Claims 1 and 27 and 2 and 28. *See, e.g., ’053 patent*, 7:51-55.

140. 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; §§IX.A.3.d-IX.A.3.e (1(b)(1)-(b)(2) and 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.

141. In my opinion, a POSITA would have understood that any server (e.g., a CDN edge cache server 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.

13. Claim 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 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.

142. In my opinion, Lewin, alone or combined with Wein teaches claims 14 and 40’s additional limitations.

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

144. 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.

145. In my opinion, 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. 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.

14. **Claim 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 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.

146. For the reasons I have explained previously, in my opinion, Lewin, alone or combined with Wein teaches claims 15 and 41 by obtaining configuration information from the NOC/registration server (*control core*). §IX.A.13 (cls. 14 and 40).

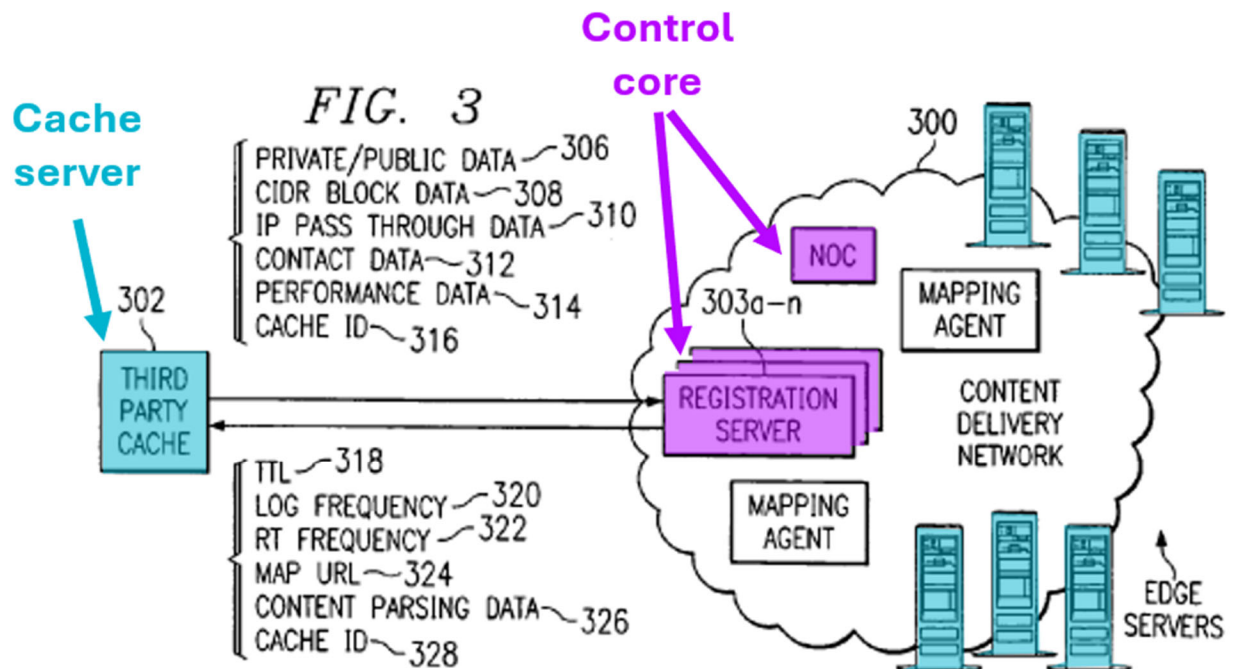
15. **Claim 16: The CDN of claim 1 wherein the control core comprises a distributed system consisting of a plurality of machines.**

Claim 42: The method of claim 27 wherein the control core comprises a distributed system consisting of a plurality of machines.

147. In my opinion, Lewin discloses or suggests claims 16 and 42's additional limitations.

148. Lewin discloses a network operations center (NOC) and 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* §IX.A.3.b (1(a) and 27(pre)(a)). In my opinion, a POSITA would have expected not only for the NOC to include a “distributed system” of machines (as claimed) to operate the variety of processes assigned to it, but also the NOC and separate registration server to be a “distributed system” because they are distinct machines that each perform a specified function and are intended to operate together to control the access and operation of the CDN.

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

149. In my opinion, Lewin discloses or suggests claim 21's additional limitations.

150. 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.

151. In my opinion, a POSITA would have understood a DNS to manage content requests constitutes a *rendezvous mechanism* consistent with the '053 patent. This is consistent with the '053 patent, which describes how "the rendezvous system 104 is used to direct client resource requests [and] is preferably implemented using the DNS." *See '053 patent*, 4:29-35.

17. **Claim 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 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:

152. In my opinion, Lewin discloses or suggests claims 22 and 47(pre)’s additional limitations.

153. The ’053 patent uses collector mechanisms to collect information—such as 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. With this understanding, 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.

154. In my opinion, a POSITA would have understood that Lewin's logging mechanism constitutes a *collector mechanism*, consistent with the '053 patent. *See '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.").

18. Claim 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 47(E): generating information; and

Claim 47(F): providing at least some of said generated information to said collector system.

155. For the reasons I have explained previously, in my opinion, Lewin discloses or suggests claims 23 and 47(E), 47(F)'s additional limitations. §IX.A.17 (cls. 22 and 47(pre)). Specifically, 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*). *Id.*

19. **Claim 49(pre): The method of claim 27, further comprising, by said particular cache server of said plurality of cache servers:**

Claim 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

Claim 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.

156. In my opinion, Lewin combined with Wein teaches claim 49.

157. In my opinion, a POSITA would have understood that CDNs will typically host content on behalf of multiple customers. 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. §§IX.A.3.d-IX.A.3.g (1(b)(1)-1(b)(2)(2) and 27(A)-27(D)).

20. **Claim 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 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.

158. For the reasons I have explained previously, Lewin alone or combined with Wein teaches claims 25 and 50's additional limitations. §§IX.A.3.f-IX.A.3.g (1(b)(2)(2)-(3) and 27(C)-27(D)). Specifically, Lewin's caches (whether alone or combined with 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.*

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

159. Liu discloses a CDN—specifically, a well-known one at the time of the '053 patent 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. *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:

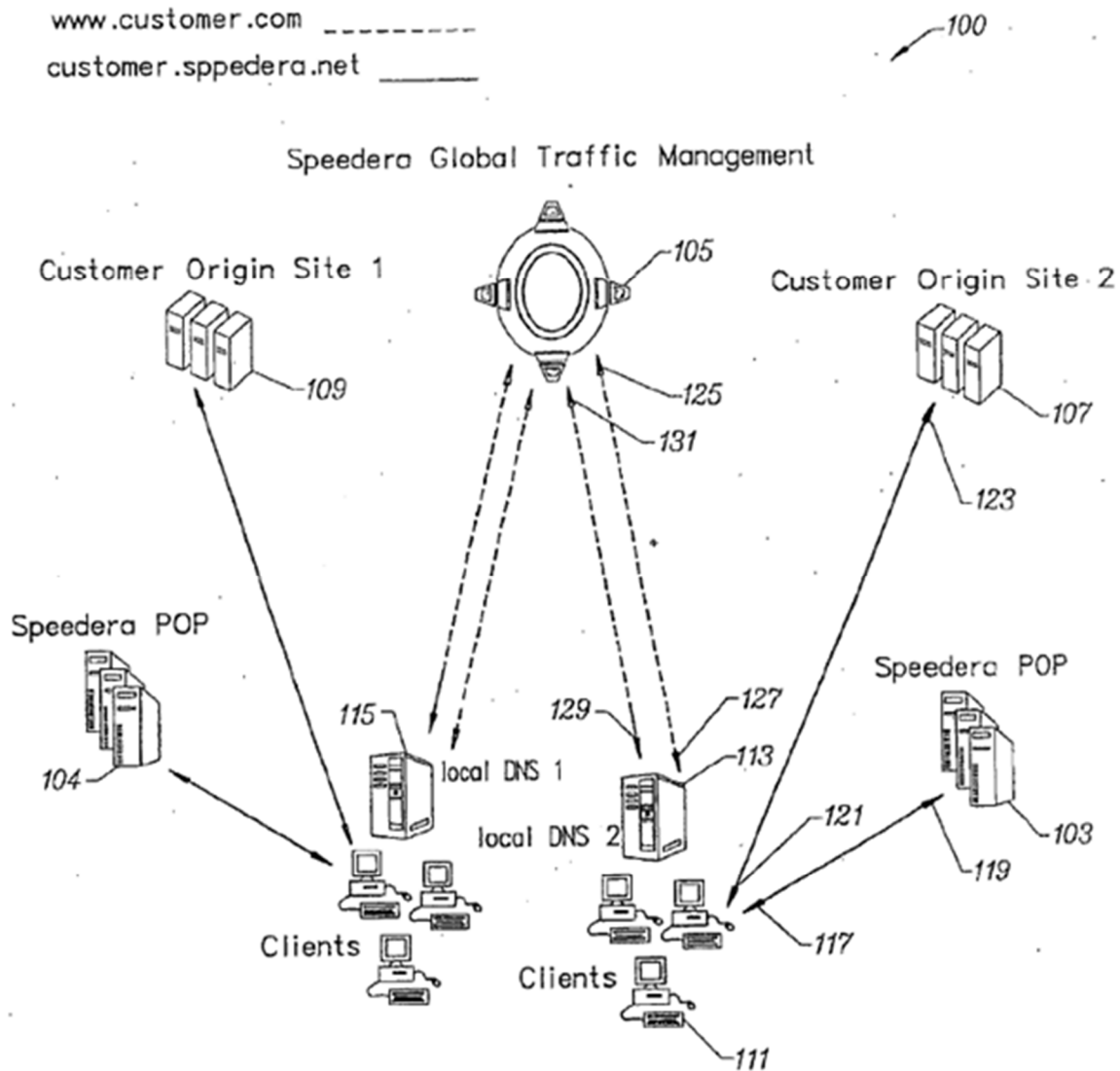


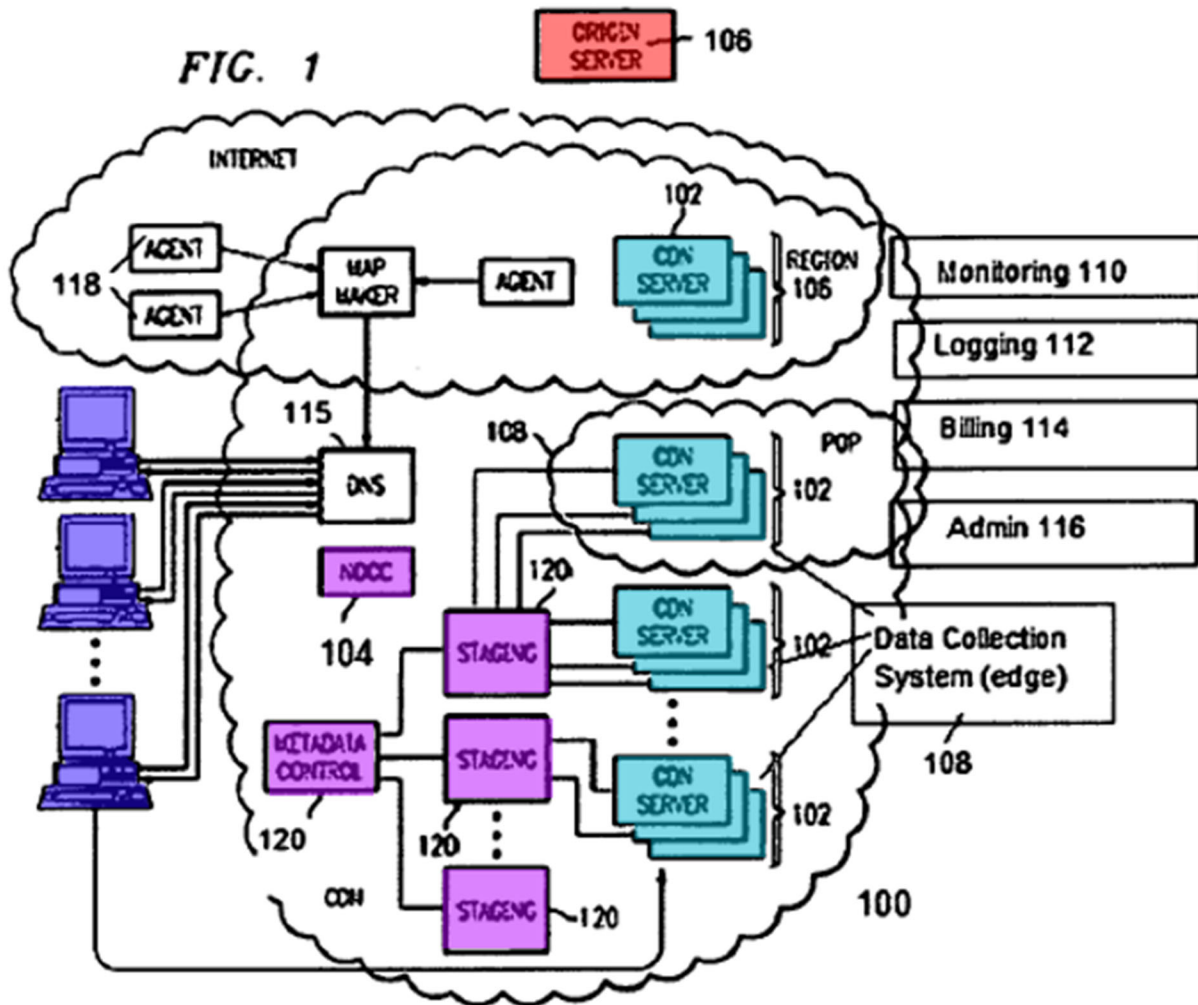
FIG. 1

Id., Fig. 1, ¶28.

Liu's UDN includes origin servers and cache servers to store content. *Id.*, ¶¶29-30, 72. 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. *Id.*, ¶¶184, 192, 205-06.

b. Devanneaux

160. Devanneaux discloses another CDN—specifically, a well-known "Akamai" CDN—that includes staging servers 102, origin server 106, edge servers 106, and a 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.

Specifically, Devanneaux configures its edge servers with metadata tags and configuration files on a “customer-specific basis.” *Id.*, ¶11.

2. Motivation to Combine Liu with Devanneaux

161. 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.

162. 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.”).

163. In my opinion, 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. 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. 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).

164. In my opinion, a POSITA would have had a reasonable expectation of success in combining Devanneaux’s added functionality 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. All

Devanneaux provides is an express disclosure of one easily implementable way to provide those cache-side configurations using eXtensible Markup Language (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. Moreover, a POSITA would have known that Speedera and Akamai networks had similar structures,⁵ and thus improvements to one were generally transferable to the other. 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 using XML. In my opinion, such an implementation would have been well within the skills of a POSITA.

⁵ Akamai merged with Speedera in 2005, around the same time as Liu and Devanneaux. *See generally Mears*.

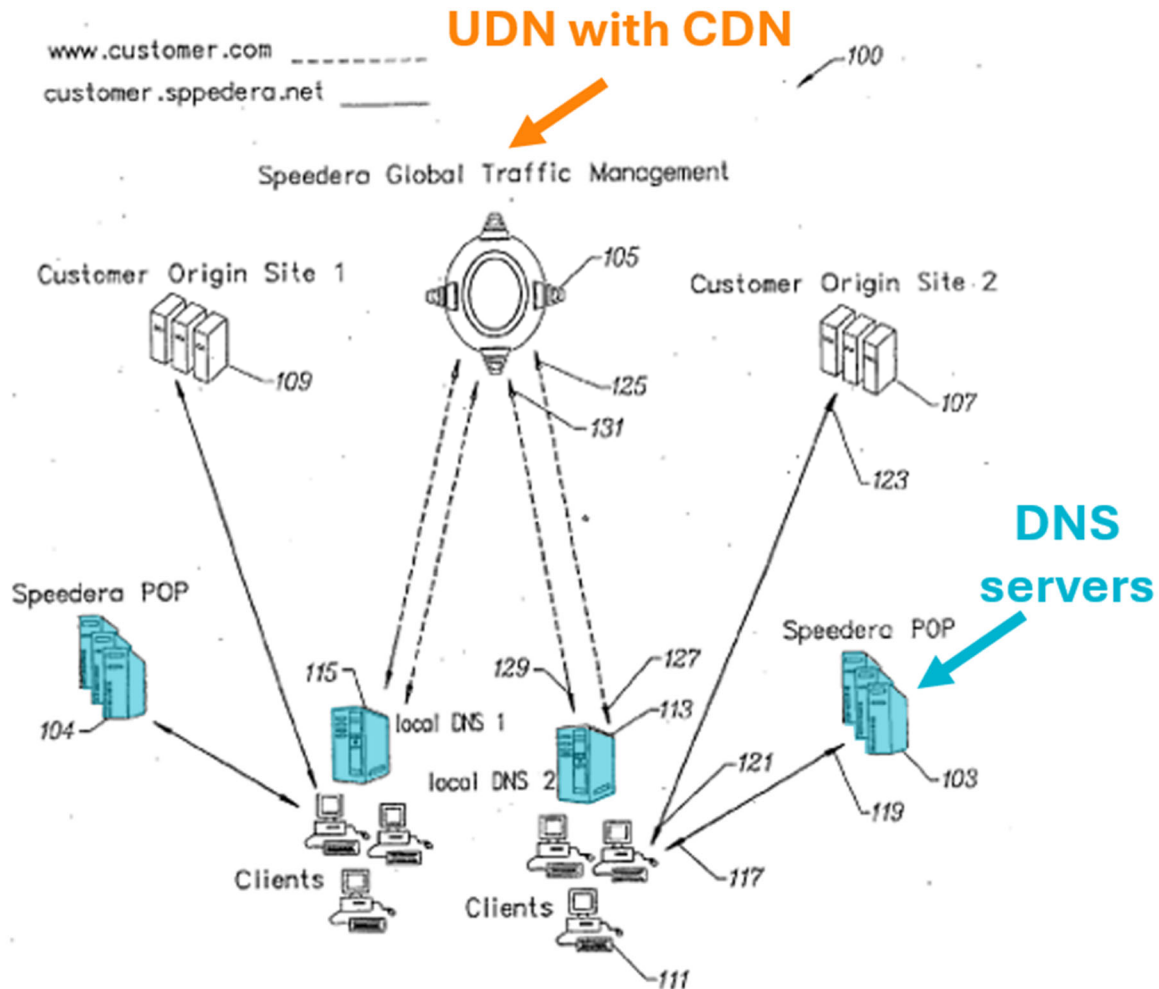
3. Claims 1, 27

a. 1(pre): A content delivery network (CDN), the CDN serving resources on behalf of one or more customers of the CDN, the CDN comprising:

27(pre): A computer-implemented method, operable in a content delivery network (CDN) comprising:

165. If the preambles are limiting, it is my opinion that Liu discloses or suggests them.

166. Liu discloses a “*computer-implemented method*” for 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 access to 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.



Id., Fig. 1, ¶¶5, 8, 28, 31-39.

167. In my opinion, a POSITA would have recognized Liu’s UDN as a CDN. Although Liu describes GTM traffic solutions, they are built into the CDN to provide more robust CDN capabilities provided as a “single outsourced solution or service to a customer” to leverage as a unified network that “eliminates the need for independent CDN and GTM solutions” and separate, expensive customer-installed hardware and software components. *Liu*, ¶¶27-29.

b. 1(a): a control core; and

27(pre)(a): a control core; and

168. In my opinion, Liu alone or combined with Devanneaux teaches or suggests 1(a) and 27(pre)(a).

169. Liu discloses that “[t]he configuration of the Speedera Network is maintained by centrally managed configuration files.” *Liu*, ¶295. Liu does not specify the exact system that “centrally manage[s]” the configuration files, but in my opinion, a POSITA would have understood that a central-management system would have one or more systems operating together to do so, which would constitute a *control core* consistent with the claims. *See ’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 that are also supposed to be “centrally managed”.

170. 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.

171. Therefore, in my opinion, a POSITA would have understood 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.

c. 1(b): a plurality of cache servers, each particular cache server of said plurality of cache servers being constructed and adapted to:

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:

172. In my opinion, Liu discloses or suggests 1(b) and 27(pre)(b).

173. 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). §IX.B.3.a (1(pre) and 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”). *Id.*, ¶¶305, 17-18; *see also id.*, ¶39 (“The steps can be performed using computer software or hardware or a combination”).

174. In my opinion, a POSITA would have 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*).

d. 1(b)(1): upon joining the CDN,

1(b)(1)(1): obtain global configuration data from the control core; and

27(A): obtaining global configuration data from the control core;

175. In my opinion, Liu discloses or suggests 1(b)(1)-1(b)(1)(1) and 27(A).

176. 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”).

177. In my opinion, a POSITA would have understood that a server first joining a network would obtain a copy of the global-configuration file (*upon joining the CDN...obtain global-configuration data*) to access, for instance, 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. Without the initial global configuration file, the server would not know how to interact with the other servers in the network because it is that configuration file that tells it how to do so. 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. §IX.B.3.b (1(a) and 27(pre)(a)).

e. **1(b)(2): having joined the CDN,**

1(b)(2)(1): obtain updated global configuration data, if needed; and

27(B): selectively obtaining updated global configuration data; and

178. In my opinion, Liu discloses or suggests 1(b)(2)-1(b)(2)(1) and 27(B).

179. 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”). In my opinion, 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]*).

180. However, if “selectively obtaining” (cl. 27) requires that the cache server *pull* updated global configuration files, in my opinion, that also would have been obvious. Pulling and pushing are two predominant ways of acquiring data across a network; pushing involves one network element sending information to a receiving network element, and pulling involves a receiving network element requesting information be sent from a sending network element. *See, e.g., Reynolds*, ¶¶132-34 (describing how push, pull, and push-pull schemes were the “three basic kinds of algorithms” to distribute information in a network); *Winter*, ¶103 (“the downloading of the server configuration is carried out in Push or Pull.”); *Wein*, 8:5-10 (“[a] metadata configuration file can be ‘pushed’ out to all servers when it changes, or ‘pulled’ periodically from a set of intermediate staging servers”). In fact, Devanneaux acknowledges that pushing and pulling are both ways to distribute at

least *content* throughout the network, which a POSITA would have understood applied more broadly to any information set, such as configuration files. *Devanneaux*, ¶37 (noting prefetching can utilize “push-to-edge” or “pull-from-edge” protocols). In my opinion, configuring Liu’s servers to pull configuration files rather than push would have amounted to a simple substitution of known elements (pulling protocols for pushing protocols), or applying one known technique (pulling) as a suitable alternative to another (pushing) to achieve the same, predictable result (updating configuration files). But a POSITA would have also been specifically motivated to configure Liu’s cache servers to not only wait for new global configuration files to be manually “pushed” using the AdminTool software, (*Liu*, ¶301), but to also provide cache servers the ability to check to see if updated configuration files should be pulled (for instance, at preset intervals or due to another triggering event). This modification would provide added flexibility to the network, since configuration updates could be scheduled based on the demands of the network. For instance, updated configuration files could first be pushed to sets of intermediate servers, and then individual edge cache servers could pull the updates at their convenience, such as during low-traffic points. And given the prevalence of both pushing and pulling techniques, swapping out one for another would have involved

minimal coding that, in my opinion, would have been well within the skills of a POSITA to implement.

f. 1(b)(2)(2): obtain customer configuration information associated with at least one customer of the CDN;

27(C): obtaining customer configuration information associated with a particular customer of the CDN;

181. In my opinion, Liu alone or combined with Devanneaux teaches or suggest 1(b)(2)(2), 27(C).

182. 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”). In my opinion, a POSITA would have understood that these controllable parameters constitute *customer-configuration information associated with [at least one/a particular customer] of the CDN* because they are tools that provide customers ways to configure what data is cached by the CDN (by providing new data, invalidating old data, or configuring how the DNS routes requests for a customer’s content to the CDN).

183. That said, in my opinion, a POSITA would have also been motivated to implement additional customer-configuration files in Liu’s network, such as those taught by Devanneaux. §IX.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. In other words, Devanneaux describes providing customers even more control over how their content is handled and served by the CDN than Liu, and provides a mechanism (using metadata files) to carry out those configurations.

184. 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, like Liu already does with the global-configuration files.

- g. 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.**

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.

185. In my opinion, Liu alone or combined with Devanneaux teaches or suggests 1(b)(2)(3) and 27(D).

186. 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). Customers may also 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.

187. In my opinion, therefore, 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.*

188. 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.

189. 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.

4. **Claim 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 28: The method of claim 27 further comprising:
(A)(2) obtaining data from one or more other locations in the CDN.**

190. In my opinion, Liu discloses or suggests claims 2 and 28's additional limitations.

191. 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.

192. In my opinion, a POSITA would have understood that, when adding a web cache or DNS cache server to the UDN (*upon joining the CDN*, cls. 2 and 28)⁶,

⁶ As I noted above, claim 2 specifies certain functions “upon [a cache server] joining the CDN,” but claim 28 is not limited to “upon joining the CDN.” Therefore, I first

the new server would need both the relevant configuration files, as well as copies of the required software and web or DNS content it is supposed to host. In my opinion, 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*). See, e.g., *Lewin*, 3:43-49 (explaining how caches are typically registered with a CDN, but “[r]egistration...may not be required, for example, if the CDN...preconfigure the cache...in advance so that it can join the CDN.”).

193. Moreover, for web cache servers, specifically, a POSITA would have understood that a new cache would need content to serve. And as I explained above, there are two ways this could happen: (1) either the cache remains empty until it receives requests, and then fetches the content from other servers in the CDN in response to requests and then caches the content requested for itself (e.g., fetch-and-

address functionality “upon joining the CDN,” which applies to both claims 2 and 28, and then functionality that also meets the claim *after* joining the CDN, which applies only to claim 28.

fill) or (2) the cache is pre-populated with content before receiving any requests. Pre-population may occur, for instance, using load-balancing servers, a central authority server, or even other peer edge servers that identify specific content for the third-party cache to store. Other prior art, for instance, would define rules in which edge servers would be pre-populated with popular content that would be expected to be in high demand. *Swildens*, ¶43 (“An edge server maintains a dynamic number of popular files in its memory for the customer...[W]hen a file has been requested from an edge server a sufficient number of times to become more popular than the lowest popular stored file, the file is obtained”), ¶130 (“popularity is based on the aggregate data from all of the edge servers that are accessing [a] common storage”).

194. A POSITA would have found it obvious, therefore, for Liu’s web caches to pre-populate its cache and obtain copies of content upon joining the CDN (*obtain data from one or more other locations in the CDN*). Moreover, a POSITA would have expected that the content would be obtained from other CDN caches (e.g., edge caches) already storing the content because a web cache would be expected to be installed within a cluster of other edge caches at some POP site that could simply transfer content between them, rather than burdening the origin server that may be located far away. *See, e.g., Seed*, ¶62 (“If a popular object has already been replicated to an edge server[,], filling a new edge server request for that object

from one of the peer edge servers may be more efficient than the parent server or origin server. Since there are typically more edge servers...there is an increased likelihood that a peer edge server may be closer in terms of network distance... [and] could also lessen the burden on the parent servers or origin servers.”).

195. 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*). *Id.*, ¶424.

5. **Claim 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 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.

196. In my opinion, Liu alone or combined with Devanneaux teaches claims 4 and 30’s additional limitations.

197. As I explained, 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. §IX.B.3.f (cls. 1(b)(2)(2) and 27(C)).

198. In my opinion, 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*).

6. Claim 6: The CDN of claim 1 wherein the customer configuration information comprises at least one CDN resource.

Claim 32: The method of claim 27 wherein the customer configuration information comprises at least one CDN resource.

199. In my opinion, Liu alone or combined with Devanneaux teaches or suggests claims 6 and 32's additional limitations.

200. As I explained, Liu alone or combined with Devanneaux teaches providing global-configuration files and customer configurations (Liu), as well as customer-configuration files (Devanneaux). §IX.B.3.f (cls. 1(b)(2)(2) and 27(C)). In my opinion, 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.

This is consistent with the '053 patent, which distinguishes between customer resources (e.g., content) and CDN resources. *See '053 patent*, 15:66-16:2.

201. 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. *See '053 patent*, 40:39-40 (noting a customer-configuration script may “provide...for CDN resources.”). For instance, customer-configuration information may specify to prioritize certain content over other content, or may request specific logging requirements associated with specific content, each of which would either implicitly or explicitly identify specific CDN resources (like priority or logging protocols) to implement.

7. Claim 9: The CDN of claim 1 wherein the global configuration data comprises at least one CDN resource.

Claim 35: The method of claim 27 wherein the global configuration data comprises at least one CDN resource.

202. In my opinion, Liu discloses or suggests claims 9 and 35's additional limitations.

203. As I explained, Liu discloses providing global-configuration information. §IX.B.3.f (cls. 1(b)(2)(2) and 27(C)). In my opinion, 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; *see* ’053 *patent*, 15:66-16:2 (distinguishing between customer resources (e.g., content) and CDN resources).

204. 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; ’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.”). For instance, Liu notes that part of the global configuration includes “the list of POPs and the servers present at each POP” as well as “the list of hostnames serviced” and “maps the hostnames to the servers that can serve the content for the hostname.” *Liu*, ¶205. A POSITA would have understood that the POP sites, servers, and hostnames are all CDN resources used to locate and serve content to clients.

8. **Claim 10: The CDN of claim 9 wherein the global configuration data comprises global configuration object (GCO).**

Claim 36: The method of claim 35 wherein the global configuration data comprises global configuration object (GCO).

205. In my opinion, Liu discloses or suggests claims 10 and 36's additional limitations.

206. 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.

207. In my opinion, a POSITA would have understood Liu's "global configuration file" both constitutes and comprises a "global configuration object" consistent with the '053 patent. The '053 patent appears to use the term "object" to refer to a single, modular unit (like a file) with the required configuration data contained within it, based on which the network acts. *See '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.”) (emphasis added), 28 (Table; defining “Partial Object” as “[a]ssembling 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”).

208. To the extent the ’053 patent requires the global-configuration object to be a discrete piece of the configuration file—like a table—used to inform content request routing, Liu teaches it. Specifically, 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).

9. **Claim 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 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.

209. In my opinion, Liu discloses or suggests claims 11 and 37's additional limitations.

210. Liu discloses distributing web content to a plurality of web cache servers, or replicating DNS content between DNS cache servers. §IX.B.4 (cls. 2 and 28).

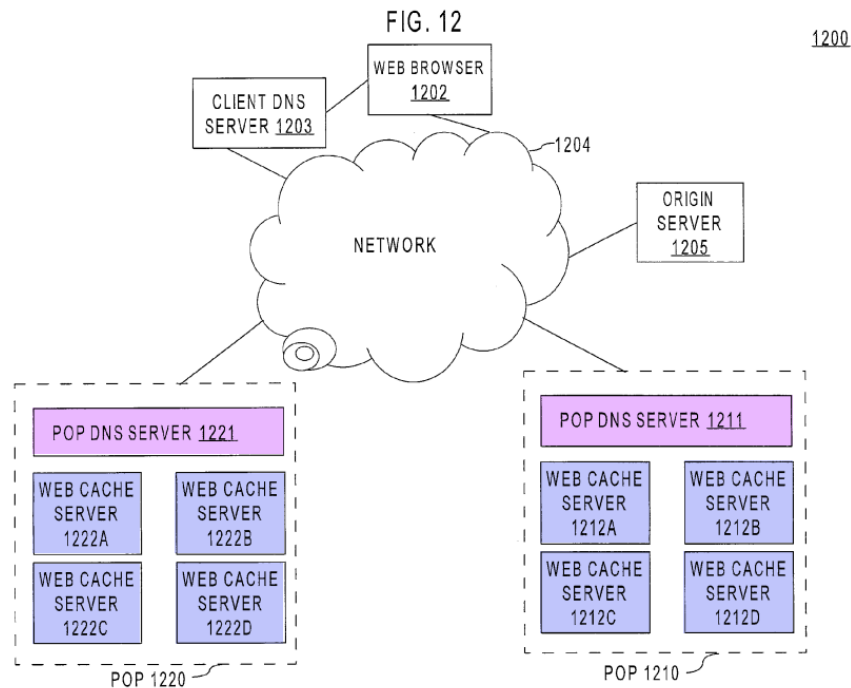
211. In my opinion, a POSITA would have understood that, upon joining a CDN (cl. 2) and/or after joining the CDN (cl. 28), both content- and DNS-type cache servers would obtain either web content or DNS content from other caches in the CDN to host the content for itself in the first instance or fetch 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). §IX.B.4 (cls. 2 and 28).

10. Claim 12: The CDN of claim 11 wherein the one or more other locations in the CDN comprise at least one peer location.

Claim 38: The method of claim 37 wherein the one or more other locations in the CDN comprise at least one peer location.

212. In my opinion, Liu discloses or suggests claims 12 and 38's additional limitations.

213. 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 (below). 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.



214. A POSITA would have considered it obvious that, upon joining a CDN (cls. 2 and 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 their peer caches in the CDN. §IX.B.4 (cls. 2 and 28); *see also, e.g.,* Seed, ¶¶20 (noting CDNs typically organized in tiers of origin, parent, and edge servers, each of which may be a “cluster” of servers), 62 (noting peer relationships between different tiers).

215. In my opinion, a POSITA would have understood that because peers are typically more numerous at the lower tier 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. *See, e.g.,* Seed, ¶62. In other words, a

POSITA would have understood caching content on peer servers (e.g., POPs) offloads that content from the origin server, which “provide[s] better quality of service (QoS) to a broad spectrum of end user systems” around the globe. *Harvell*, Fig. 1, ¶16.

11. Claim 13: The CDN of claim 2 wherein the particular cache server determines said one or more other locations using said information⁷ in said global configuration data.

Claim 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.

216. In my opinion, Liu discloses or suggests claims 13 and 39’s additional limitations.

217. 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; §IX.B.3.d-IX.B.3.e (1(b)(1)-(b)(2) and 27(A)-(B)).

218. In my opinion, a POSITA would have understood that, upon joining the CDN (cls. 2 and 28), the server would use the information provided in the configuration file to locate resources—whether additional software or content—

⁷ See *supra* note 4.

stored elsewhere in the network to install/load onto the server. §IX.B.4 (cls. 2 and 28). Indeed, identifying the locations of resources is (at least in part) one of the primary purposes of the global-configuration file.

219. 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.

12. Claim 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 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.

220. In my opinion, *Liu* discloses or suggests claims 14 and 40’s additional limitations.

221. As I explained, *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*). §§IX.B.3.d-IX.B.3.e (1(b)(1)-1(b)(2) and 27(A)-(B)).

13. **Claim 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 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.

222. In my opinion, Liu discloses or suggests claims 15 and 41's additional limitations.

223. As I explained, in my opinion, a POSITA would have understood that the original and updated "centrally managed" global-configuration files would be obtained from the NOC (*control core*). §§IX.B.3.d-IX.B.3.e (1(b)(1)-1(b)(2) and 27(A)-(B)).

14. **Claim 16: The CDN of claim 1 wherein the control core comprises a distributed system consisting of a plurality of machines.**

Claim 42: The method of claim 27 wherein the control core comprises a distributed system consisting of a plurality of machines.

224. In my opinion, Liu discloses or suggests claims 16 and 42's additional limitations.

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

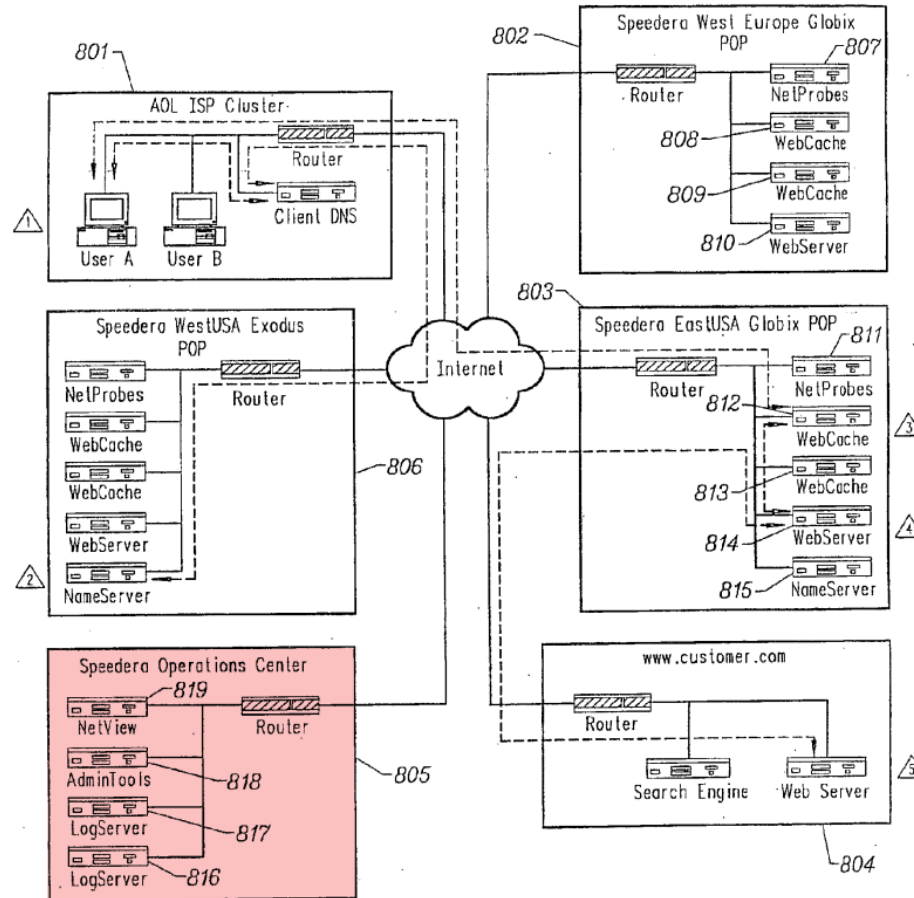


FIG. 8

Liu, Fig. 8, ¶¶17, 263-64.

226. In my opinion, a POSITA would have understood these machines constitute a “distributed system,” as claimed, because they are distinct machines that

each perform a specified function and are intended to operate together as part of a centralized operations center.

15. **Claim 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 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.

227. In my opinion, Liu discloses or suggests claims 18 and 44's additional limitations.

228. 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.

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

230. In my opinion, 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. *Id.*, ¶434 (“[E]dgeDNS cache servers can be deployed at any number of POPs[and] configured to serve multiple domains as well as multiple customers”).

16. **Claim 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 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.

231. In my opinion, Liu-Devanneaux teaches or suggests claims 19 and 45.

232. 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.

233. In my opinion, 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). §IX.B.3.f (cls. 1(b)(2)(2) and 27(C)); *Nygren*, 12 (describing cache control and indexing as a customer-controlled and metadata-implemented configuration).

17. Claim 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 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.

234. In my opinion, Liu discloses or suggests claims 20 and 46’s additional limitation.

235. Liu discloses that the cache periodically checks the validity of its content. *Liu*, ¶81; §IX.B.16 (cls. 19 and 45). Liu suggests such checks are at

predetermined time intervals, but notes that “[t]his interval can be configured depending upon ones needs.” *Liu*, ¶81.

236. In my opinion, 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. For instance, a POSITA would have recognized that limiting checks to times when the CDN receives a request for content 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.

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

237. In my opinion, Liu discloses or suggests claim 21’s additional limitations.

238. 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).

239. In my opinion, a POSITA would have understood a DNS to manage content requests constitutes a *rendezvous mechanism* consistent with the ’053 patent.

This is consistent with the '053 patent, which describes how “the rendezvous system 104 is used to direct client resource requests [and] is preferably implemented using the DNS.” *See '053 patent*, 4:29-35.

19. Claim 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 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:

240. In my opinion, Liu discloses or suggests claims 22 and 47(pre)'s additional limitations.

241. 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.

242. In my opinion, a POSITA would have understood that the Log Server constitutes a *collector mechanism* consistent with the '053 patent. 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.”).

20. Claim 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 47(E): generating information; and

Claim 47(F): providing at least some of said generated information to said collector system.

243. In my opinion, Liu discloses or suggests claims 23 and 47(E)-(F)’s additional limitations.

244. 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. A POSITA would have understood that a “daemon” is a background

task that provides services without direct user interaction⁸ and Liu describes many different daemons. *See, e.g., Liu*, ¶¶56 (Log Distributor, Log Collector daemons), 207 (Service Probe daemon), 211 (Latency Probe daemon), 330 (Log Server daemon). 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.

21. Claim 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 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.

245. In my opinion, Liu discloses or suggests claims 24 and 48's additional limitations.

246. 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.

⁸ *See, e.g., What is a Daemon?* by Lenovo, available at:

<<https://www.lenovo.com/us/en/glossary/what-is-a-daemon/>>.

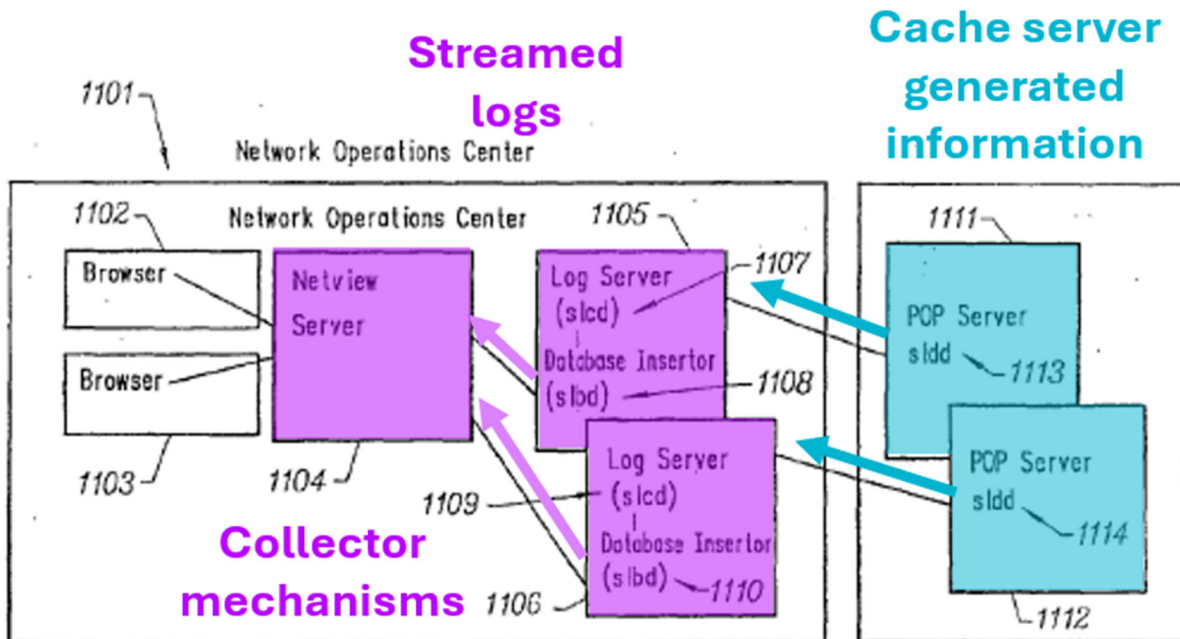


FIG. 11

Id., Fig. 11.

247. In my opinion, to the extent transmitting intermittent log files does not constitute “streaming,” a POSITA would have understood that there are only two time-based ways to transmit data from one network element to another: (1) transmit entire files (or portions of complete files) at once; or (2) stream data in real time. *Id.*, ¶55 (noting media may be “a live streaming event” compared to “large file downloads”).

248. In my opinion, 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). *See 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).

249. A POSITA, however, 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. 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.

22. **Claim 49: The method of claim 27, further comprising, by said particular cache server of said plurality of cache servers:**

Claim 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

Claim 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.

250. In my opinion, Liu-Devanneaux teaches claim 49.

251. As I explained, 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. §§IX.B.3.d-IX.B.3.g (1(b)(1)-1(b)(2)(3) and 27(A)-27(D)).

23. **Claim 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 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.

252. For the reasons I have previously provided, in my opinion, Liu-Devanneaux teaches or suggests claims 25 and 50. §IX.A.3.f-IX.A.3.g (cls. 1(b)(2)(2)-(3) and 27(C)-(D)). Specifically, 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. *Id.*

24. **Claim 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.**

253. In my opinion, Liu-Devanneaux teaches or suggests claim 26.

254. 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).

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

1. Prior Art

a. Nygren

255. Nygren provides an overview of one of the most well-known CDNs offered by a company called Akamai. *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; and explains how the network requires

“[c]onfiguration changes and software updates” that need to be rolled out efficiently.

Id., 6; *see also id.* at 14-15.

256. Like Liu and Devanneaux, Nygren explains that “[m]any Akamai system components need to receive frequent configuration updates,” including “customer metadata configuration files used to configure the edge server platform.” *Id.*, 13-14; *see also id.*, 5-6 (providing “a configuration management platform that allows enterprise customers to retain fine-grained control how their content and applications are served to the end user”).

257. Nygren explains that, circa 2010, internet services required “rigorous end-to-end system quality enterprise applications” because “even small degradations in performance reliability can have a considerable business impact,” e.g., losing a single hour of customer access to an e-commerce site can cost a business hundreds of thousands to millions of dollars in revenue because, in part, customers that must wait more than 3 seconds for a page to load abandon a site (and therefore a sale). *Id.*, 2. Drivers of Internet degradation include latency (delay), packet loss, network outages, inefficient protocols, and inter-network friction. *Id.*, 3-4.

258. Nygren explains that Akamai Technologies specifically “introduced the Content Delivery Network (CDN) concept to address these challenges” circa 2000. *Id.*, 4; *see also id.*, 2. Nygren explains that the Akamai CDN first addressed

elementary challenges with global Internet traffic bottlenecks by caching site content close to end users (at the “edge of the Internet”), then matured into a holistic solution that could handle full, virtual application delivery networks built over the Internet to deliver HD-quality live and on-demand media content, execute entire Java J2EE⁹ applications, and enhance network visibility and control with new tools (e.g., a management portal, reporting and analytics, security, action logging). *Id.*, 4-5. The Akamai platform translates a user-requested URL into the IP address of one server of a 60,000 machine global network—an edge server at the periphery of the network and most proximate to a content-requesting user—to serve that content by downloading any uncacheable content from an origin server and managing network traffic using a communications and control system, data collection and analysis system, and management portal:

⁹ Java 2 Platform, Enterprise Edition, a set of specifications and APIs used for large-scale, multi-tier enterprise applications. *See, e.g., Java 2 Platform, Enterprise Edition (J2EE) Overview* by Oracle, available at: <<https://www.oracle.com/java/technologies/appmodel.html>>.

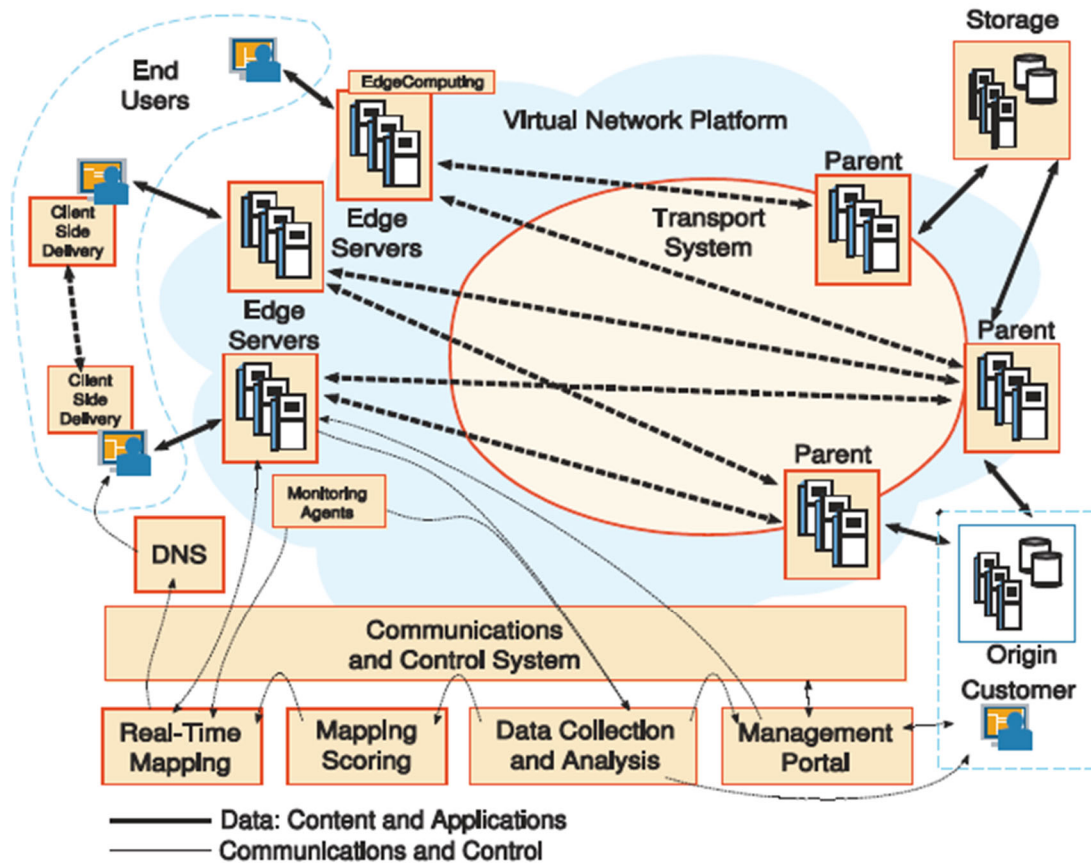


Figure 5: System components of the Akamai platform.

Id., Fig. 5, 12-14.

2. **Claim 17:** The CDN of claim 16 wherein the control core uses a distributed consensus algorithm to achieve consensus among the plurality of machines.

Claim 43: The method of claim 42 wherein the control core uses a distributed consensus algorithm to achieve needed consensus among the plurality of machines.

259. In my opinion, the combination of Liu, Devanneaux, and Nygren teaches or suggests claims 17 and 43.

260. 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.

261. Nygren, however, taught a protocol called “PAXOS” and “decentralized leader election” (*consensus algorithms*), which Akamai’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.

262. 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*). Indeed, the ’053 patent itself envisioned using PAXOS as its control-core consensus algorithms for this exact purpose. *See ’053 patent*, 18:40-20:13.

263. In my opinion, 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 CDN. In fact, the '053 patent acknowledged that numerous prior-art and commercially available PAXOS protocols were known and implemented in distributed networking systems. *See, e.g., '053 patent*, 19:42-20:9. In my opinion, implementing commercially available protocols like PAXOS for their intended purpose would have been well within the skills of a POSITA.

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

1. Prior Art

a. Lewin819

264. Lewin819 is another reference that discloses implementation details of Akamai's CDNs. *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. *Lewin819*, 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):

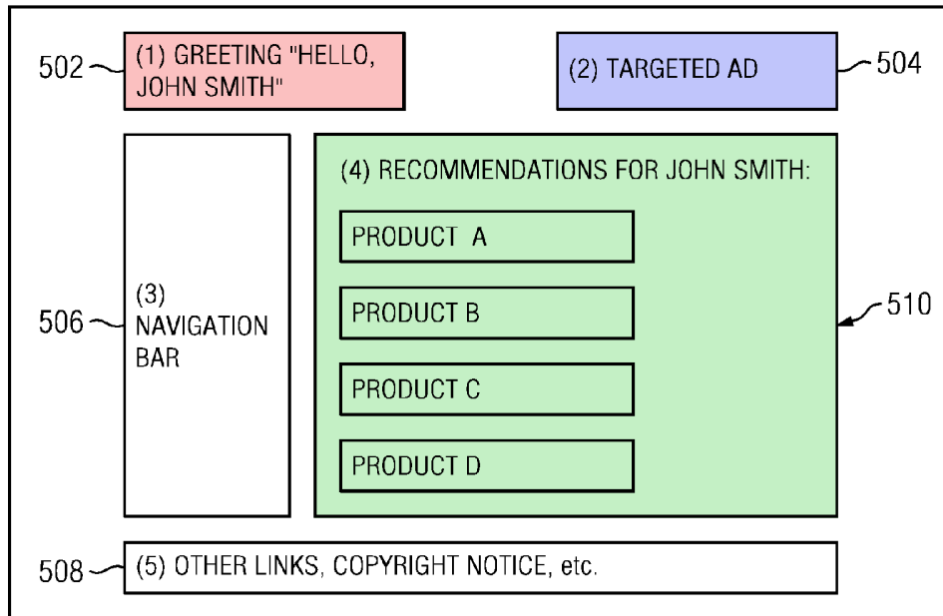


FIG. 5

Id., Fig. 5. The associated scripts are shown below as an html file that contains multiple ESIs that are described by “[p]referably, the content provider leverages an “edge side include” (ESI) markup language that is used to define Web page fragments for dynamic assembly at the edge.” *Id.*, Abstract. Thus, a POSITA would have understood that the statements below that begin with “<esi.” include statements for specific Web page fragments that the server is to use to dynamically build the Web page for the response.

EX-1003, PTAB Case No. IPR2026-00190
Declaration of Mark Lanning in support of Petition for
Inter Partes Review of U.S. Patent No. 9,456,053 B2

```
<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.

2. Motivation to Combine with Lewin819

265. 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 2010) 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.

266. In my opinion, 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*. §IX.D.1.a (*Lewin819* Summary). 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.

267. A POSITA would have also 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. Moreover, Lewin819 provides the necessary framework to carry out these dynamic webpages and scripts in a CDN environment like Liu's. *Lewin819*, ¶¶34-36; *see also, e.g.*, the ESI Language Specification 1.0¹⁰, a standard explaining that ESI logic uses ESI scripts and metadata tags to assemble and deliver cached content.

3. **Claim 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 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.

268. In my opinion, the combination of Liu, Devanneaux, and Lewin819 teaches or suggests claims 3 and 29.

¹⁰ *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>>.

269. Liu and Devanneaux both teach customer-configuration information/files that are programmed by the customer and then distributed to servers in the CDN. §IX.B.3.f (Ground 2A, 1(b)((2)(2) and 27(C)). However, Liu and Devanneaux do not expressly discuss a server obtaining customer-configuration information upon receipt of a request for a resource associated with the customer.

270. Upon implementing a CDN capable of building dynamic webpages using customer-provided metadata (§IX.D.2 (motivation)), in my opinion, a POSITA would have 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). Lewin819, for instance, 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*." *Id.*, ¶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.

4. **Claim 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 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.

271. In my opinion, the combination of Liu, Devanneaux, and Lewin819 teaches or suggests claims 5 and 31.

272. Liu and Devanneaux both teach customer-configuration information/files that are programmed by the customer and then distributed to servers in the CDN. §IX.B.3.f (Ground 2A, 1(b)(2)(2) and 27(C)). Devanneaux also teaches the ability to at least *parse* configuration information from metadata files by reading scripts. *Devanneaux*, Fig. 5, ¶¶18, 28-29. However, Liu and Devanneaux do not expressly discuss a server in which the customer-configuration information itself *is* a script.

273. Upon implementing a CDN capable of building dynamic webpages using customer-provided metadata (§IX.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.* 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.

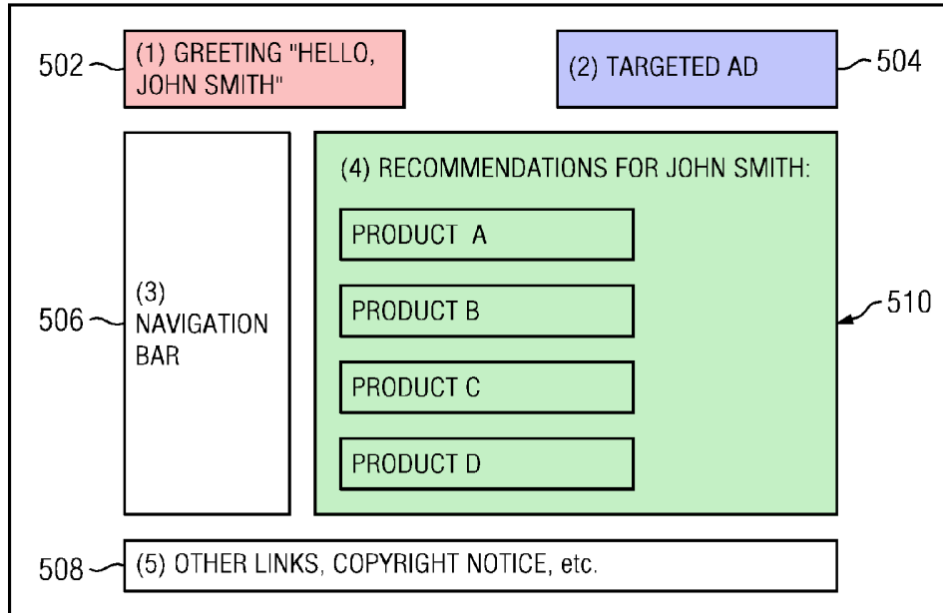


FIG. 5

EX-1003, PTAB Case No. IPR2026-00190
Declaration of Mark Lanning in support of Petition for
Inter Partes Review of U.S. Patent No. 9,456,053 B2

```
<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.

274. Additionally, 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. *See, e.g., Cuan*, ¶26, which describes the use of scripts in configuration files to enable custom processes, such as notifying customers of successful content deliveries.

- 5. Claim 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 33: The method of claim 27 wherein the customer configuration information comprises at least one customer configuration script (CCS) associated with the particular customer.

275. For the reasons I have previously provided, in my opinion, the combination of Liu, Devanneaux, and Lewin819 teaches or suggests claims 7 and 33. Specifically, the Liu-Devanneaux-Lewin819 combination teaches that the customer configuration information comprises customer-provided scripts used by the CDN to build dynamic websites (*customer-configuration script*) for the same reasons as described above. §IX.D.4 (cls. 5 and 31).

6. **Claim 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 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.

276. In my opinion, the combination of Liu, Devanneaux, and Lewin819 teaches or suggests claims 8 and 34.

277. As I explained, (§IX.D.3 (cls. 3 and 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).

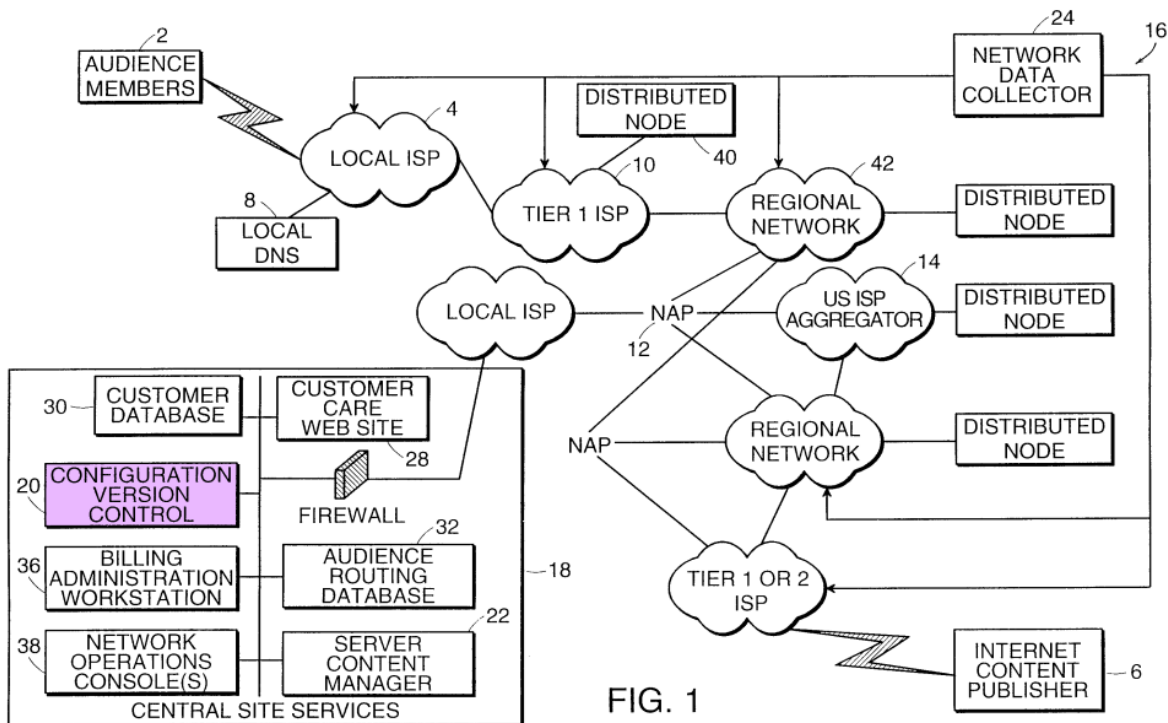
E. Ground 2D: Obviousness over Liu, Devanneaux, and Crosbie (Claims 18-20, 44-46)

1. Prior Art

a. Crosbie

278. 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.



2. Motivation to Combine with Crosbie

279. 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.

280. In my opinion, 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. Crosbie, for instance, provides one example version control-scheme. §IX.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. In my opinion, 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).

281. A POSITA would have also had a reasonable expectation of success in making this modification because, in my opinion, 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. 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.

3. **Claim 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 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.

282. In my opinion, Liu-Devanneaux-Crosbie teaches or suggests claims 18 and 44.

283. Liu discloses cache servers operate in accordance with global configuration files that can be updated as needed. §IX.B.3.e (1(b)(2)-1(b)(2)(1) and 27(B)).

284. Upon implementing a centralized version control module in Liu's UDN (§IX.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 because Liu's cache servers already query other servers for content updates (*Liu*, ¶ 435) and were therefore predisposed to query the Configuration Version Control server that runs the centralized version module because it "acts as a point of control and audit for the rebroadcaster (*cache server*) configuration files" that Crosbie's cache servers run. *Crosbie*, 17:30-31. Also, such version-checking mechanisms were extremely old and well-known in the art. *See, e.g., 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*). *Crosbie*, 10:31-11:9.

4. **Claim 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 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.

285. In my opinion, Liu-Devanneaux-Crosbie teaches or suggests claims 19 and 45.

286. 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. §IX.B.3.f (1(b)(2)(2) and 27(C)).

287. Upon implementing a centralized version control module in Liu's UDN (§IX.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 because Liu's cache servers already query other servers for content updates (*Liu*, ¶ 435) and were therefore predisposed to query the Configuration Version Control server that runs the centralized version module because it "acts as a point of control and audit for the rebroadcaster (*cache server*) configuration files" that Crosbie's cache servers run. *Crosbie*, 17:30-31. Such version-checking mechanisms were extremely old and well-known in the art. *See, e.g., Lee*, ¶51; *Lee* 617, ¶¶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*). *Crosbie*, 10:31-11:9.

5. **Claim 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 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.

288. In my opinion, Liu-Devanneaux-Crosbie teaches or suggests claims 20 and 46.

289. 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, *see* §IX.B.3.e-IX.B.3.f (1(b)(2)-(b)(2)(2) and 27(B)-(C)); (2) automatically at predetermined time intervals or specified times (*see, e.g., Lee*, ¶51); or (3) in response to a triggering event (*Crosbie*, 11:2-9, describing version rollback “in the event of a misconfiguration”). Choosing any of these options would have been straightforward, with each having tradeoffs: autonomous versioning may ensure accuracy as the risk of tying up servers or propagating problems with that version (*Dilley*, 55), but manual versioning may risk running old, outdated configurations no longer pertinent

to serving a customer's dynamic and constantly changing content delivery needs (*id.*).

290. A POSITA would have recognized that a trigger received by a server may be a request for content, and it 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 going to be used in servicing the request while balancing CDN resources. *See, e.g., Liskov*, 6:45-67 (providing a replication status report of content cached on a server in response to a caching server query); *Lee*, ¶51 (as an alternative to automatic versioning, asking a user when and whether to download a newer version “when convenient”); *Davis*, Fig. 11, 14:8-19 (“hot swapping” technique for handing off existing requests initiated on an old content delivery application version to an updated application version by handing over new requests to the new version and phasing out the old as a function of the *requests*).

X. SUMMARY AND ADDITIONAL REMARKS

291. It is my opinion that Grounds 1 and 2A-D render the Challenged Claims unpatentable as obvious, as I explained in the paragraphs preceding.

292. I currently hold the opinions expressed in this declaration based on the information available to me at this time. I may, however, form additional or new

EX-1003, PTAB Case No. IPR2026-00190
Declaration of Mark Lanning in support of Petition for
Inter Partes Review of U.S. Patent No. 9,456,053 B2

opinions should additional information or insights be made available, or upon further consideration.