Trials@uspto.gov
571-272-7822

Paper No. 33
Entered: May 14, 2015

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

RELOADED GAMES, INC.,
Petitioner,

v.

PARALLEL NETWORKS LLC,
Patent Owner.

Case IPR2014-00136
Patent 7,188,145 B2

Before KRISTEN L. DROESCH, BRIAN J. McNAMARA, and HYUN J. JUNG, *Administrative Patent Judges.*

JUNG, *Administrative Patent Judge*.

FINAL WRITTEN DECISION
35 U.S.C. § 318(a) and 37 C.F.R. § 42.73

I. INTRODUCTION

Reloaded Games, Inc. ("Petitioner") filed a Petition (Paper 4, "Pet.") on November 11, 2013, requesting institution of an *inter partes* review of claims 1–36 of U.S. Patent No. 7,188,145 B2 ("the '145 patent") pursuant to 35 U.S.C. §§ 311–19. Parallel Networks LLC ("Patent Owner") filed a Preliminary Response. Paper 9. Based on these submissions, we instituted *inter partes* review of claims 2–4, 6, 7, 10, 16–18, 20, 21, 24, and 29–36 under 35 U.S.C. § 103. Paper 15 ("Dec. on Inst.").

After institution, Patent Owner filed a Patent Owner's Response (Paper 23, "PO Resp."), and Petitioner filed a Reply (Paper 24, "Reply"). In addition, the parties rely upon expert testimony. Petitioner proffered the Declaration of Dr. Peter B. Danzig (Ex. 1002, "Danzig Declaration") with its Petition. Patent Owner proffered the Declaration of Dr. Mitchell A. Thornton (Ex. 2002, "Thornton Declaration"). No deposition transcripts were filed, and no motions were filed by the parties.

A combined oral hearing in this proceeding and Case IPR2014-00139 was held on February 23, 2015, and a transcript of the hearing is included in the record (Paper 32, "Tr.").

We have jurisdiction under 35 U.S.C. § 6(c). This final written decision is issued pursuant to 35 U.S.C. § 318(a) and 37 C.F.R. § 42.73. For the reasons that follow, we determine that Petitioner has not shown by a preponderance of the evidence that claims 2–4, 6, 7, 10, 16–18, 20, 21, 24, and 29–36 of the '145 patent are unpatentable.

A. The ’145 Patent (Ex. 1001)

The ’145 patent is titled “Method and System for Dynamic Distributed Data Caching” and issued March 6, 2007. The ’145 patent issued from application 09/759,406, which was filed on January 12, 2001.

Reproduced below is Figure 6 of the ’145 patent.

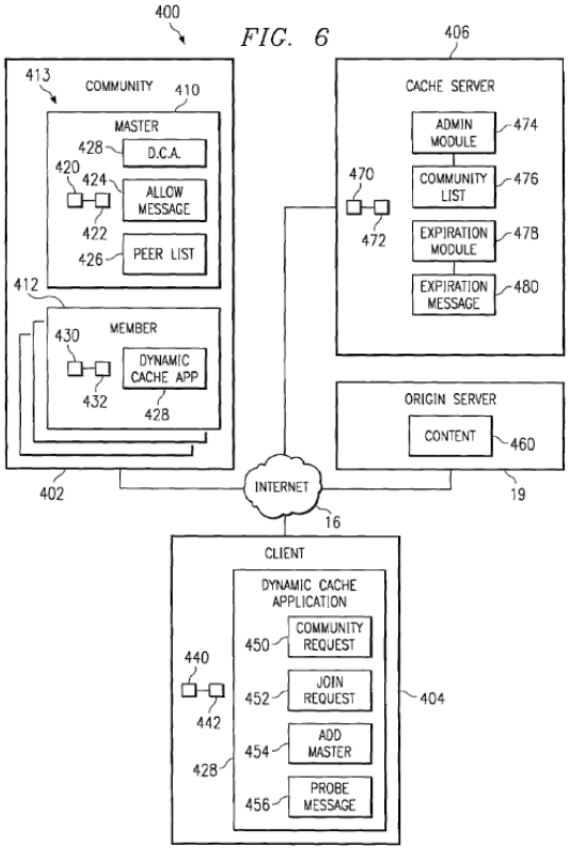

Figure 6 depicts a block diagram illustrating a dynamic caching system according to one embodiment. Ex. 1001, 4:56–58. Community 402 comprises one or more peers 413, and peers 413 further comprise master 410 and member 412. *Id*. at 17:60–63. Each peer 413 includes dynamic cache application 428, which provides functionality to support distributed caching system 10. *Id*. at 18:1–3. Client 404 comprises a computer also executing dynamic cache application 428 that is operable to generate join request 452,

which is a data message indicating that client 404 wishes to join a particular community 402. *Id*. at 18:66–67, 19:14–15, 21–22. Master 410 is operable to generate allow message 424 that comprises a data message sent to client 404 to inform client 404 either that it is being allowed to join community 402 or that entry to community 402 is denied. *Id*. at 18:22–27.

In operation, dynamic cache application 428 of client 404 generates community request 450, which is a request for a list of communities 402 that client 404 may attempt to join. Ex. 1001, 20:19–23; *see also id*. at 23:43–46 (describing a method for adding client 404 to community 402), Fig. 9. Community request 450 is communicated to cache server 406. *Id*. at 20:23–24; *see id*. at 23:44–46. After selecting a particular community 402, dynamic cache application 428 of client 404 generates join request 452, which is communicated to master 410 of community 402. *Id*. at 20:41–48; *see id*. at 23:46–24:9. After receiving join request 452, master 410 determines whether to allow client 404 to become a member 412 of community 402 by use of a suitable criterion, such as whether the addition of client 404 would exceed the maximum number of members 412 for community 402 or whether the round trip transit time for data between client 404 and present members 412 is within a certain threshold. *Id*. at 20:49–58; *see also id*. at 24:65–25:8 (describing a method for allowing client 404 to join community 402), Fig. 10. If master 410 determines that client 404 can be a member, dynamic cache application 428 at master 410 generates allow message 424, which then joins client 404 to community 402. *Id*. at 20:64–21:6; *see id*. at 25:9–10, 17–21. If master 410 determines that client 404 should not join community 402, then dynamic cache application 428 at

master 410 generates allow message 424 indicating that client 404 has been denied entry to community 402, or may ignore join request 452 so that client 404 determines that it has been denied entry. *Id*. at 21:14–21; *see id*. at 25:10–16.

Once client 404 is allowed to join community 402, master 410 updates peer list 426 to include client 404, and communicates the updated peer list 426 to members 412 to inform them that client 404 has joined community 402. Ex. 1001, 21:7–9; *see id*. at 25:21–30. Dynamic cache application 428 then reallocates content 460 to be cached among master 410, members 412, and client 404. *Id*. at 21:10–13.

B. Illustrative Claims

The '145 patent has 36 claims, of which claims 2–4, 6, 7, 10, 16–18, 20, 21, 24, and 29–36 are being challenged. Claims 29, 32, 35, and 36 are independent. Claim 29 is a method claim, and claims 32, 35, and 36 are system claims. Claim 2, its base claim 1, and claim 29 are illustrative and reproduced below.

> 1. A method for dynamic distributed data caching comprising:
>
> providing a cache community on a first side of a point of presence, the cache community comprising at least one peer, the cache community being associated with content obtained from a second side of the point of presence, the content being cached by the at least one peer;
>
> allowing a client to join the cache community;
>
> updating a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community;
>
> associating the content with the client based on joinder of the client;

re-allocating the cache storage of the content among the peers in the cache community in response to allowing the client to join the community.

2. The method for dynamic distributed data caching according to claim 1 and further comprising:

receiving a join request from the client; and

determining whether to allow the client to join the cache community.

29. A method for dynamic distributed data caching comprising:

communicating a community request to an administration module;

receiving a community list from the administration module in response to the community request, the community list including a list of communities;

selecting one of the communities to attempt to join;

generating a join request to attempt to join the selected one of the communities;

receiving an allow message associated with the selected one of the communities;

receiving a peer list associated with the selected one of the communities;

receiving content allocated for storage in caches of peers in the peer list for cache storage re-allocation in response to joining the selected one of the communities; and

providing content for cache storage re-allocation to peers in the peer list in response to joining the selected one of the communities.

C. Asserted Ground of Unpatentability

We instituted the instant *inter partes* review on the ground that, under 35 U.S.C. § 103, claims 2–4, 6, 7, 10, 16–18, 20, 21, 24, and 29–36 would have been obvious over Smith (U.S. Patent No. 6,341,311 B1, issued

January 22, 2002 (Ex. 1006)) and Inohara (U.S. Patent No. 6,256,747 B1, issued July 3, 2001 (Ex. 1007)). Dec. to Inst. 40.

II. ANALYSIS

A. Claim Construction

In an *inter partes* review, "[a] claim in an unexpired patent shall be given its broadest reasonable construction in light of the specification of the patent in which it appears." 37 C.F.R. § 42.100(b); *see In re Cuozzo Speed Techs., LLC*, 778 F.3d 1271, 1279–83 (Fed. Cir. 2015). There is a "heavy presumption" that a claim term carries its ordinary and customary meaning. *CCS Fitness, Inc. v. Brunswick Corp.*, 288 F.3d 1359, 1366 (Fed. Cir. 2002); *In re Translogic Tech., Inc.*, 504 F.3d 1249, 1257 (Fed. Cir. 2007).

1. Previously Interpreted Terms

In the Decision on Institution, we interpreted various claim terms of the '145 patent as follows:

Term	**Interpretation**
"means for providing a cache community on a first side of a point of presence, the cache community comprising at least one peer, the cache community being associated with content obtained from a second side of the point of presence, the content being cached by the at least one peer"	Function: "providing a cache community on a first side of a point of presence, the cache community comprising at least one peer, the cache community being associated with content obtained from a second side of the point of presence, the content being cached by the at least one peer" Structure: "one or more general purpose computers programmed to create a new community"

Term	Interpretation
"means for allowing a client to join the cache community"	Function: "allowing a client to join the cache community" Structure: "one or more general purpose computers programmed to evaluate a join request to determine whether the client will be allowed to join the cache community based on a criterion and decide whether the client is allowed to join the cache community based on the evaluation"
"means for updating a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community"	Function: "updating a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community" Structure: "one or more general purpose computers"
"means for associating the content with the client based on joinder of the client"	Function: "associating the content with the client based on joinder of the client" Structure: "one or more general purpose computers programmed to update an allocation list table to include the client"
"means for re-allocating the cache storage of the content among the peers in the cache community in response to allowing the client to join the community"	Function: "re-allocating the cache storage of the content among the peers in the cache community in response to allowing the client to join the community" Structure: "one or more general purpose computers programmed to renegotiate cache shares among peers in the cache community and update an allocation list table to reflect which peers cache which content"

Term	Interpretation
"means for communicating a community request to an administration module"	Function: "communicating a community request to an administration module" Structure: "an Internet connection that is always available"
"means for receiving a community list from the administration module in response to the community request, the community list including a list of communities"	Function: "receiving a community list from the administration module in response to the community request, the community list including a list of communities" Structure: "software or hardware associated with the client operably connected to the Internet for receiving a community list"
"means for selecting one of the communities to attempt to join"	Function: "selecting one of the communities to attempt to join" Structure: "one or more general purpose computers programmed to evaluate various factors associated with the communities on the community list to determine which community the client should join"
"means for generating a join request to attempt to join the selected one of the communities"	Function: "generating a join request to attempt to join the selected one of the communities" Structure: "software, hardware, or software and hardware associated with the client operable to provide a data message over the Internet, which indicates that the client wishes to join the selected one of the communities"

Term	Interpretation
"means for receiving an allow message associated with the selected one of the communities"	Function: "receiving an allow message associated with the selected one of the communities" Structure: "software, hardware, or software and hardware associated with the client operable to receive a data message over the Internet, which indicates to the client that the client is being allowed to join the selected one of the communities"
"means for receiving a peer list associated with the selected one of the communities"	Function: "receiving a peer list associated with the selected one of the communities" Structure: "software, hardware, or software and hardware associated with the client operable to receive a data message over the Internet, which indicates a list of peers in the selected one of the communities"
"means for receiving content allocated for storage in caches of peers in the peer list for cache storage re-allocation in response to joining the selected one of the communities"	Function: "receiving content allocated for storage in caches of peers in the peer list for cache storage re-allocation in response to joining the selected one of the communities" Structure: "software or hardware associated with each of the peers in the peer list operable to receive content for storage in cache"

Term	Interpretation
"means for providing content for cache storage re-allocation to peers in the peer list in response to joining the selected one of the communities"	Function: "providing content for cache storage re-allocation to peers in the peer list in response to joining the selected one of the communities" Structure: "software, hardware, or software and hardware associated with each of the peers in the peer list or an origin server, each operable to provide content for cache storage to peers in the peer list"
"community"	"similarity or identity" or "sharing, participation, and fellowship"
"allow" or "allowing"	"to permit the presence of"

See Dec. on Inst. 7–14.

Patent Owner does not argue against our construction above and instead provides arguments based on our construction in its Response. PO Resp. 2, 14, 18; Tr. 36:22–37:10. Petitioner also uses our construction above in replying to Patent Owner. Reply 1, 2, 6–7. Based on the complete record before us, we see no reason to change our original construction of the above terms.

B. Asserted Ground under 35 U.S.C. § 103

Petitioner argues that claims 2–4, 6, 7, 10, 16–18, 20, 21, 24, and 29–36 would have been obvious over Smith and Inohara by referring to disclosures in the references, a claim chart, and the Danzig Declaration. Pet. 34–51.

1. Smith (Ex. 1006)

Smith describes a method, computer program product, and system for routing URL data object requests in a proxy server array and involves an array of multiple proxy servers configured to act together as a single distributed cache. Ex. 1006, Abstract, 1:9–10.

Reproduced below is Figure 5 of Smith.

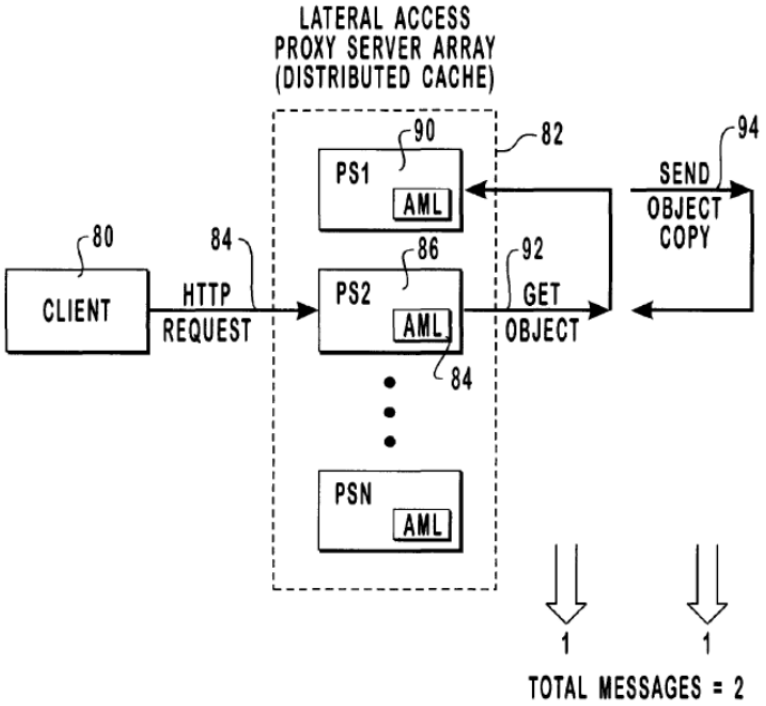

FIG. 5

Figure 5 of Smith shows proxy server array 82 that allows lateral access of a URL object. Ex. 1006, 6:5–11, 9:59–63. Client 80 sends HTTP request 84 for a URL object to proxy server 86 of proxy server array 82. *Id*. at 9:63–65. Proxy server 86 has an array membership list that contains all the proxy servers of array 82. *Id*. at 9:65–67. Proxy server 86 uses the membership list to determine which server in array 82 should contain the

requested URL object. *Id*. at 10:1–5. In one embodiment, proxy server 86 uses a deterministic hashing function to hash the URL object of request 84 and combines it with a deterministic hash of each server name on the membership list to find the server most likely to have the requested URL object. *Id*. at 10:5–22. Thereafter, proxy server 86 directs a get object request 92 to the server most likely to have the requested URL object, such as server 90, and server 90 responds by sending a copy of the requested URL object to server 86, which, in turn, responds to HTTP request 84 from client 80. *Id*. at 10:23–31.

Smith states that "many different implementations may be envisioned by those skilled in the art that will allow a proxy server to be added to the proxy server array." Ex. 1006, 18:51–53. Smith states that "[a]fter beginning at step 194, a new proxy server is designated as being added to the array at step 196." *Id*. at 18:54–55, Fig. 11, steps 194, 196.

2. Inohara (Ex. 1007)

Inohara relates to a plurality of computers connected by a network to distribute and share information. Ex. 1007, 1:8–12.

Reproduced below is Figure 1 of Inohara.

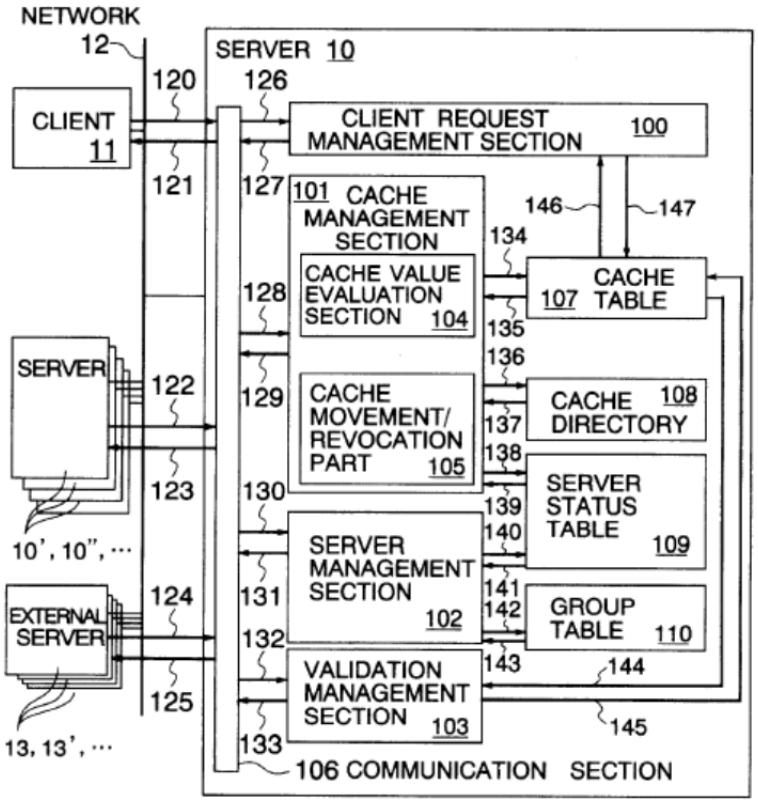

Figure 1 of Inohara depicts a block diagram of Inohara's invention. Ex. 1007, 5:7–8. A user uses client 11 to obtain information from server 10 or external server 13 through network 12. *Id*. at 5:38–41. Server 10 caches the information for client 11 obtained from external server 13. *Id*. at 5:41–46.

Server 10 has data structures that include cache table 107, cache directory 108, server status table 109, and group table 110. Ex. 1007, 6:13–15. Cache table 107 provides a cache of information. *Id*. at 7:11–22. Cache directory 108 lists which server 10 holds the cached information. *Id*. at 7:23–30. Server status table 109 holds the IDs of other servers 10 and attributes of these servers 10, such as throughput and latency. *Id*. at 7:31–

45. Group table 110 holds proximate server groups selected from servers on the server status table 109. *Id*. at 7:46–48. A group of servers has a maximum number of servers or members. *Id*. at 7:48–49. A server in the group of servers having the smallest server ID is a leader. *Id*. at 7:51–52. The leader acts as a relay for communication between groups. *Id*. at 7:54–56.

Hierarchy formation begins when server 10 is started, and server 10 refers to server status table 109. Ex. 1007, 9:17–19, 24–41, 10:31–32. Server 10 selects a server from server status table 109 and transmits a message requesting "group table transfer." *Id*. at 9:42–46. Once the selected server receives the "group table transfer" request message, the selected server sends a message with its group table 110. *Id*. at 9:48–52. Server 10 updates its server status table 109 with the received group table 110 and measures communication speed between server 10 and the selected server based on the response time to its "group table transfer" request message. *Id*. at 9:46–48, 52–62. Server 10 completes similar steps with the remaining servers on its server status table 109. *Id*. at 10:9–13. Server 10 then transmits group participation message 300 to the most proximate server in a group of servers on server status table 109. *Id*. at 10:13–17. When server 10 receives group update message 320 in response to group participation message 300, server 10 transmits a "group table transfer" request message to a leader of group update message 320. *Id*. at 10:19–23. After the leader receives the "group table transfer" request message, the leader transmits a message with its group table 110 to server 10. *Id*. at 10:24–27.

Figure 5 of Inohara is reproduced below.

FIG. 5

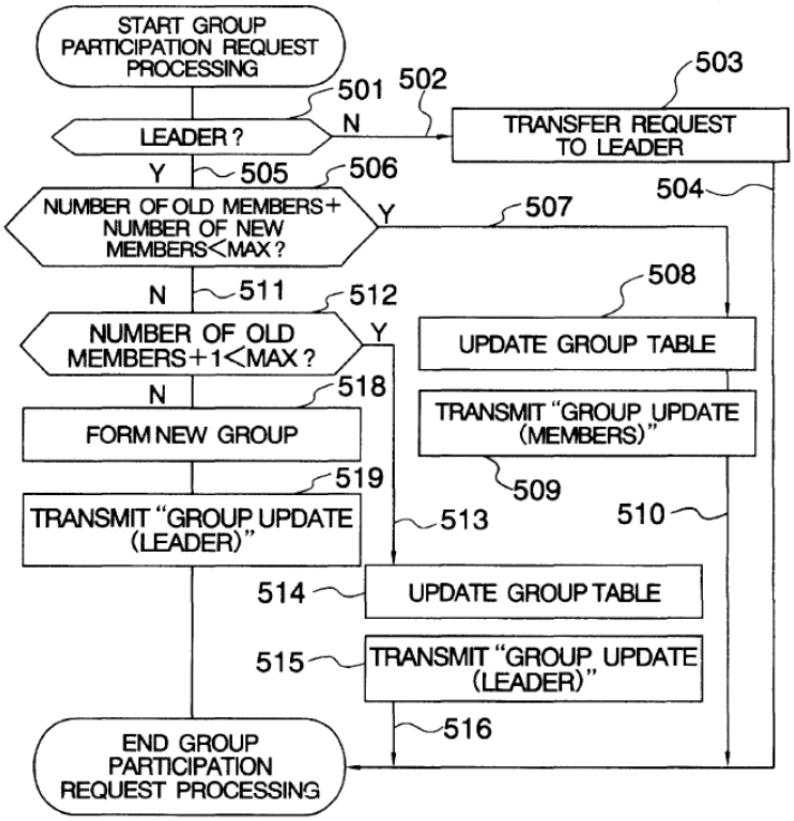

Figure 5 shows a flow chart of group participation processing. Ex. 1007, 5:14. When a leader of a group (server 10 or the server having the smallest server ID on group table 110) receives group participation request message 300, the leader examines its group table 110 to determine the number of old members and examines group participation message 300 to determine the number of new members. *Id*. at 7:46–52, 10:41–60, Fig. 5 (steps 501–505). Server 10 then determines if the number of old and new members is smaller than a maximum number. *Id*. at 10:60–63. If the number of old and new members is smaller than the maximum number, then

servers in group participation message 300 are added to group table 110. *Id.* at 10:63–66, Fig. 5 (steps 506–508). Group table 110, thus, would include the group leader, the old members, and the new members, and group participation request processing would be complete. *Id.* at 10:66–11:5.

If the number of old and new members is not smaller than the maximum number, then a judgment is made whether the addition of one of the new members would be smaller than the maximum number. Ex. 1007, 11:6–8, Fig. 5 (steps 511–512). If so, then one of the new members is added to group table 110 so that group table 110 includes the leader, the old members, and the one new member. *Id.* at 11:8–13, Fig. 5 (steps 513–514). The newly added member becomes the leader of the remaining new members through a group update message sent to the remaining new members, and group participation request processing is completed. *Id.* at 11:13–17, Fig. 5 (steps 515–516).

If the addition of even one of the new members would exceed the maximum number for a group, then the leader becomes a leader for one of the old members and one of the new members through group update message 320. Ex. 1007, 11:18–21, Fig. 5 (steps 517–519). Subsequent group update messages 320 instruct the remaining old members that their new leader is the one old member still under the leader and the remaining new members that their new leader is the one new member now under the leader. *Id.* at 11:21–37.

3. Analysis

To prevail in its challenges to the patentability of the claims, Petitioner must prove unpatentability by a preponderance of the evidence. 35 U.S.C. § 316(e); 37 C.F.R. § 42.1(d).

A claim is unpatentable under 35 U.S.C. § 103(a) if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). To establish obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *See CFMT, Inc. v. Yieldup Int'l Corp.*, 349 F.3d 1333, 1342 (Fed. Cir. 2003); *In re Royka*, 490 F.2d 981, 985 (CCPA 1974).

A patent claim composed of several elements, however, is not proved obvious merely by demonstrating that each of its elements was known, independently, in the prior art. *KSR*, 550 U.S. at 418. In that regard, for an obviousness analysis it is important to identify a reason that would have prompted one of skill in the art to combine prior art elements in the way the claimed invention does. *Id.* A precise teaching directed to the specific subject matter of a challenged claim, however, is not necessary to establish obviousness. *Id*. Rather, obviousness must be gauged in view of common sense and the creativity of an ordinarily skilled artisan. *Id*. Moreover, obviousness can be established when the prior art, itself, would have suggested the claimed subject matter to a person of ordinary skill in the art. *In re Rinehart*, 531 F.2d 1048, 1051 (CCPA 1976).

a. Level of Skill in the Art

In determining the level of skill in the art, various factors may be considered, including "type of problems encountered in the art; prior art solutions to those problems; rapidity with which innovations are made;

sophistication of the technology; and educational level of active workers in the field." *In re GPAC, Inc.*, 57 F.3d 1573, 1579 (Fed. Cir. 1995) (citing *Custom Accessories, Inc. v. Jeffrey-Allan Indus., Inc.*, 807 F.2d 955, 962 (Fed. Cir. 1986)). There is evidence in the record before us that reflects the knowledge level of a person with ordinary skill in the art. Petitioner's Declarant, Mr. Danzig, attests that a person with ordinary skill in the art would be an individual who possesses "a B.S. in computer science or related engineering discipline or equivalent experience and at least two years in networking or equivalent experience or education" and "some knowledge of networking computers, distributed systems, data caching, and implementation of distributed networks in computer systems." Ex. 1002 ¶ 9. Patent Owner does not dispute Mr. Danzig's assessment of the level of ordinary skill in the art.

b. Claims 2–4, 6, 7, and 10

Claims 2–4, 6, 7, and 10 depend from claim 1 and necessarily include "allowing a client to join the cache community," as recited by claim 1. Petitioner asserted that claims 1, 8, 9, 11–15, 22, 23, and 25–28 are anticipated by Smith. Pet. 25–34. However, we determined in the Decision on Institution that Petitioner did not show that Smith discloses "allowing a client to join the cache community," and, thus, Petitioner did not show a reasonable likelihood of succeeding on its contention that Smith anticipates claims 1, 8, 9, 11–15, 22, 23, and 25–28. Dec. on Inst. 23–24.

Claim 2 recites "[t]he method for dynamic distributed data caching according to claim 1 and further comprising: receiving a join request from the client and determining whether to allow the client to join the cache

community." Petitioner argues that "Inohara discloses that the leader of a group receives a 'group participation message' (i.e., join request) from a server (i.e., client) that wants to participate in the group" (Pet. 36 (citing Ex. 1007, 8:1–8, 10:13–17, 10:38–54, Figs. 4, 5)) and that "the leader determines whether to allow the server to join the group based on whether the group has reached a maximum number of members" (Pet. 36–37 (citing Ex. 1007, 10:38–11:3)). *See* Pet. 37–39 (citing Ex. 1007, 6:18–24, 7:58–61, 8:24–29, 10:19–30, 10:66–11:3, Figs. 2, 5).

Petitioner contends that one skilled in the art would have combined Smith and Inohara "to include the function of allowing proxy servers the ability to search for and join arrays" and that "such a modification would increase the effectiveness and performance of the system described in Smith due to the resulting large-scale cache that extends over a plurality of servers." Pet. 35 (citing Ex. 1002 ¶ 17). Petitioner asserts that the "skilled artisan would have . . . appreciated that this improvement to Smith could be achieved simply by adding the functionality of allowing proxy servers to request a listing of arrays and join an array through submitting a request to join" and "[s]uch a modification would have yielded predictable results without requiring undue experimentation." *Id*.

Patent Owner responds that "neither Smith nor Inohara teaches the limitation of '*allow[ing] a client to join the cache community,*' a limitation upon which each of claims 2–4, 6, 7, 10, 16–18, 20, 21, and 24 depend." PO Resp. 1. Patent Owner argues that Petitioner's assertion that a server group of Inohara is a cache community is incorrect because "Inohara actually teaches that servers are always added to the hierarchy" and "does not teach

that a single server group is a cache community." *Id.* at 14–16 (citing Ex. 3:65–4:9, 4:23–28; Ex. 2002 ¶ 44).

Specifically Patent Owner argues that Inohara teaches reorganizing server groups "such that communications would be more efficient if the server group were split into sub-groups" and "never discusses denying a new server or telling a new server to go find another smaller group." PO Resp. 19; *see id.* at 5–6 (citing Ex. 1007, 1:41–44, 1:46–49,3:16–22, 3:29–31, 3:48–50, 4:20–22, 7:58–64, 9:17–10:36, Figs. 2, 4; Ex. 2002 ¶ 29), 8–9 (citing 1007, 11:6–42, Fig. 5; Ex. 2002 ¶¶ 29, 32–33); Tr. 38:16–40:13, 41:1–19. Patent Owner also argues that "Inohara teaches that in all cases the new servers are admitted into the server group" and "sub-groups are not distinct and separate groupings." PO Resp. 19–20 (citing Ex. 2002 ¶¶ 37–40, 43–44); *see id.* at 9–11 (citing Ex. 1007, 10:60–66, 11:6–42; Ex. 2002 ¶¶ 34–35, 38, 39, 42–44); Tr. 42:6–47:7. Patent Owner, thus, argues that Inohara's reorganizing of a group "is not 'allowing' with the possibility of denial or a determination of entrance into the server group." PO Resp. 20.

Patent Owner further argues that "[e]ven if Inohara could be said to teach the denial of entry of a server into an intended server group, a single server group is not a cache community" and thus, "could not be construed as the denial of entry into a cache community, because . . . the server would still be placed into the overall cache hierarchy." PO Resp. 20–21 (citing Ex. 1007, 11:18–42; Ex. 2002 ¶ 44); Tr. 42:6–47:7. Patent Owner asserts that "any server that is included elsewhere in the hierarchy outside of a particular server group (because it was purportedly 'denied' entry to the server group) cannot be said to be 'denied' entrance into the cache community . . . because

the server and the particular server group still work in concert to cache data as one hierarchical community." PO Resp. 23.

Patent Owner also argues that, consistent with the construction of "community," the overall hierarchy of Inohara, not the server groups making up the hierarchy, is a community that shares, participates, and forms a fellowship in caching content. PO Resp. 16 (citing Ex. 1007, 7:58–64, Fig. 2). Patent Owner argues that a server group "stores only a portion of the content cached by the hierarchy," "could not be operated as a cache community by itself because it would not provide access to all requests for cached content without accessing the complete hierarchy," and shares additional content with servers outside the group. PO Resp. 16–17 (citing Ex. 1007, 6:48–53, 17:1–19, 17:46–48, 8:14–24); *see id*. at 6–7 (citing Ex. 1007, 4:4–9, 4:23–32, 6:48–53, 8:14–24, 11:39–42, 17:1–19, 17:46–48; Ex. 2002 ¶ 29). Patent Owner contends that the "process of forming the groups . . . is executed for the purpose of building out the described hierarchy into a single cache community" and "an individual server group is merely a building block of the hierarchical cache structure that forms the cache community of Inohara." PO Resp. 17–18 (citing Ex. 1007, 11:38–42; Ex. 2002 ¶ 44); *see id.* at 7–8 (citing Ex. 1007, 4:5–6, 7:46–64, 11:39–42, 14:21–33, 14:46–53, Fig. 5). Patent Owner, thus, argues that finding that the server group of Inohara is a community would be inconsistent with our construction of community. PO Resp. 17 (citing Ex. 1007, 4:23–32; Ex. 2001 ¶¶ 41–42).

Petitioner replies that the "individual server groups of Inohara are each a 'community'" and the server group of Inohara has characteristics that

satisfy the interpretation of "community." Reply 3 (citing Ex. 1007, 7:47–49, 7:59–61, 8:1–2, 24–26, 10:19–30, 10:51–11:17, 11:32–37); Tr. 15:15–17:2. Petitioner argues that "when a server is not permitted to join a group, 'a new group is formed.'" *Id.* (citing Ex. 1007, 11:8–19). Petitioner also argues Inohara discloses that a cache directory is shared among members of a group but does not disclose that the cache directory is shared with other groups. *Id.* at 4 (citing Ex. 1007, 15:19–21, 17:49–55, 18:24–27). Petitioner further argues that the interpretation of "community" does not require membership in a single community. *Id.* at 4–5. Petitioner further contends Inohara discloses that, on startup, servers initially form one group, thus meeting Patent Owner's narrower interpretation of "community." *Id.* at 5–6 (citing Ex. 1007, 9:17–41; PO Resp. 16).

Petitioner also argues that "[w]hen the leader of a group determines that the group is at maximum capacity, instead of allowing a new client to join the group, the client requesting to participate in the group must form a new group." Reply 7 (citing Ex. 1007, 11:18–37, 11:39–42), 8–9 (citing Ex. 1007, 7:31–64, 10:13–19, 11:18–37, 19:33–20:19); Tr. 13:19–15:14. Petitioner thus argues that "[n]ot allowing a client into the group is denying its ability to join." *Id.* Petitioner further argues that the '145 patent describes a "nearly identical process." Reply 7–8 (citing Ex. 1001, 13:33–36, 19:33–20:10, 20:53–55, 24:23–45, Fig. 9); Tr. 13:19–15:14.

We agree with Petitioner that not allowing a client into a particular group of Inohara is denying the client from joining the group. We interpreted "allowing" as "permitting the presence of" (Dec. on Inst. 13–15), and we find that Inohara teaches permitting the presence of one or more

servers into one of its server groups (*see* Ex. 1007, 10:41–11:37) consistent with our interpretation.

We also interpreted "community" as "similarity or identity" or "sharing, participation, and fellowship" (Dec. on Inst. 12–13). Based on the complete record before us, we find that Inohara's server group has "similarity or identity" or "sharing, participation, and fellowship" pertaining to managing communications as the number of servers grows. For example, Inohara discloses that a group has a maximum number of servers and each group has a leader that acts as a relay for communications between groups. Ex. 1007, 7:48–56. Inohara also discloses that "[b]y limiting, server information to be propagated, to information of proximate servers and remote servers not larger in number than a fixed number and making the number of destinations of propagation not larger than a fixed number in accordance with the structure of a multi-cast hierarchy, communication for management does not explode even if the number of servers becomes large." Ex. 1007, 14:27–33. Inohara further discloses that a "multi-cast hierarchy includes groups of a tree structure formed by server groups . . . composed of several to several-tens servers." Ex. 1007, 4:4–7. We, thus, determine that Inohara groups its servers to manage communications as more servers are added to its multi-cast hierarchy.

We also find that Inohara's multi-cast hierarchy has "similarity or identity" or "sharing, participation, and fellowship" pertaining to providing a cache. Inohara describes the problem of exchanging cache lists between servers in a plurality of distributed servers so that information absent in one cache can be queried in another cache in the plurality of distributed servers.

Ex. 1007, 3:41–46. Inohara addresses this problem by providing a protocol that "performs the propagation of a cache directory (a list of URL's and servers which hold the URL's in caches) using the multi-cast hierarchy." Ex. 1007, 4:23–32. Also, Inohara describes "a distributed server environment in which servers can join groups and share the contents of their caches through the exchange of a cache directory." *See* Pet. 34 (citing Ex. 1007, 1:8–15, 4:23–32, 9:16–10:36). However, Inohara does not disclose limiting the propagation of the cache directory to within a server group that forms the multi-cast hierarchy. Therefore, we determine the multi-cast hierarchy of Inohara to be the "community" because the servers of the hierarchy have "similarity or identity" and "sharing, participation, and fellowship" pertaining to cache provision.

Accordingly, consistent with our interpretations of "allowing" and "community" and our findings from Inohara, if a server is not allowed to join a first group, the server still participates in providing a cache in a second group but cannot participate in managing communications with the first group. As pointed out by Petitioner, on startup, servers would be in a single group initially, and the hierarchy would only have one group. Reply 5–6. Before reaching the maximum number of servers for the group, each additional server would be added to the group and thus, the hierarchy. *See* Ex. 1007, 9:16–11:5. However, when the addition of one more server would exceed the maximum number of servers for the first group, then that server would not be allowed to join the first group but still would participate in providing a cache as part of the hierarchy. *Id.* at 4:4–9, 11:18–42. In other words, the similarity or identity the additional server would have with the

servers of the first group is that it provides a cache, and the distinction is that the additional server does not manage communications with the servers of the first group. Thus, we agree with Patent Owner that the "community" of Inohara includes servers not allowed to join the first group but still provide a cache. Even on startup, we determine that Inohara's hierarchy (first group and one server not allowed to join) is the relevant "community" for our analysis. Therefore, because Inohara does not teach or suggest allowing a server to join its hierarchy, we conclude that Petitioner has not shown by a preponderance of the evidence that Inohara teaches or suggests "determining whether to allow the client to join the cache community," as recited by claim 2 and "allowing a client to join the cache community," as recited by claim 1, from which claims 2–4, 6, 7, and 10 depend. Also, based on our findings from Inohara, even if Smith were modified with the teachings of Inohara as asserted by Petitioner (Pet. 35), Petitioner's asserted combination does not teach or suggest the requirement of allowing a client to join a cache community.

Accordingly, based on the complete record before us, we determine that Petitioner has not shown by a preponderance of the evidence that, under 35 U.S.C. § 103, claims 2–4, 6, 7, and 10 are unpatentable over Smith and Inohara.

c. Claims 16–18, 20, 21, 24, and 35

Claims 16–18, 20, 21, and 24 depend from claim 15 and necessarily include "allow a client to join the cache community," as recited by claim 15. Petitioner relies on "Smith in view of Inohara as applied to the corresponding limitations" that appear in these claims. Pet. 39–40 (citing

Ex. 1007, 5:64–6:2). Patent Owner responds that Smith and Inohara fail to teach or suggest, *inter alia*, the limitation "allow a client to join the cache community," as recited by claim 15, for the reasons argued for claims 2–4, 6, 7, and 10. PO Resp. 25–26.

For independent claim 35, which recites "[a] system for dynamic distributed data caching," Petitioner argues that Smith and Inohara disclose "means for allowing a client to join the cache community." Pet. 46–47 (citing Ex. 1006, 18:51–53; Ex. 1007, 5:64–6:2, 10:38–11:3, Fig. 5). Relying on its arguments for claims 2–4, 6, 7, and 10, Patent Owner responds that Smith and Inohara fail to teach or suggest that limitation. PO Resp. 29.

For the reasons discussed above for claims 2–4, 6, 7, and 10, and based on the complete record before us, we determine that Petitioner has not shown by a preponderance of the evidence that, under 35 U.S.C. § 103, claims 16–18, 20, 21, 24, and 35 are unpatentable over Smith and Inohara.

d. Claims 29–34 and 36

Independent claim 29 recites "[a] method for dynamic distributed data caching." For the step of "receiving a community list from the administration module in response to the community request, the community list including a list of communities," Petitioner contends that "Inohara teaches that a newly started server receives a group table (i.e., community list) in response to a group table transfer request message (i.e., community request)" and that the "table includes information identifying multiple server groups." Pet. 40–41 (citing Ex. 1007, 7:46–48, 9:48–54, Figs. 2, 4). For "selecting one of the communities to attempt to join," Petitioner argues that

"Inohara discloses that the newly started server selects a group to join by measuring communication speed to proximate servers, selecting the most proximate server, and joining that server's group." *Id*. at 41–42 (citing Ex. 1007, 7:32–41, 8:55–10:17, Figs. 2, 4). For independent claims 32 and 36, Petitioner relies on Smith and Inohara as applied to claim 29. *Id*. at 45 (citing Ex. 1006, 8:25–94; Ex. 1007, 5:64–6:2), 48–50 (citing Ex. 1006, 7:5–18, 11:26–38, 11:52–55, 12:42–13:24, 17:9–18, 18:49–19:4, Figs. 6, 11; Ex. 1007, 5:64–6:2, 6:3–6, 6:10–13, 6:54–7:8, 7:58–61, 8:1–8, 8:24–29, 8:55–10:17, 9:42–46, 9:48–53, 10:19–30, 10:66–11:4, Figs. 2, 4).

Patent Owner responds that "[s]ince Inohara describes a way to implement a single cache community using a hierarchical tree structure, the reference cannot be read to disclose *selecting between multiple cache communities*" and "does not disclose multiple cache communities to select from." PO Resp. 27 (citing Ex. 2002 ¶¶ 42–44). Patent Owner, thus, argues that "the transfer of a group table . . . cannot be interpreted to teach or suggest 'receiving a community list . . . , *the community list including a list of communities*' or '*selecting one of the communities to attempt to join*.'" *Id.* Patent Owner also argues that "the process of Inohara results in a server being *assigned* to a group or location in a hierarchy, not in a server *selecting* a cache community to join from a list of cache communities." *Id.*

Petitioner replies that Patent Owner misconstrues "community" in arguing that Inohara does not teach or suggest a list of communities, for the reasons Petitioner asserted for claims 2–4, 6, 7, 10, 16–18, 20, 21, 24, 25, and 35. Reply 12–13. Petitioner also argues that "a joining server in Inohara sends a group participation message to 'the most proximate server in

a group of servers,'" that "is determined as the server with the maximum value of the division of throughput by latency" and "not a random assignment of a server." *Id.* at 13 (citing Ex. 1007, 10:13–17). Petitioner, thus, argues that Inohara's process for selecting the most proximate server is similar to the '145 patent's preferred embodiment. *Id.* (citing Ex. 1001, 23:54–62, 19:58–20:7; Ex. 1007, 10:13–17).

For the reasons discussed above for claims 2–4, 6, 7, and 10, Patent Owner's arguments that Inohara's cache community is its overall hierarchy of servers in groups are persuasive. Thus, we conclude that Petitioner has not shown by a preponderance of the evidence that Smith and Inohara teach or suggest a community list including a list of communities and selecting one of the communities to attempt to join, as required by claims 29–34 and 36.

Accordingly, based on the complete record before us, we determine that Petitioner has not shown by a preponderance of the evidence that, under 35 U.S.C. § 103, claims 29–34 and 36 are unpatentable over Smith and Inohara.

III. CONCLUSION

Petitioner has not demonstrated, by a preponderance of the evidence, that, under 35 U.S.C. § 103, claims 2–4, 6, 7, 10, 16–18, 20, 21, 24, and 29–36 of the '145 patent are unpatentable over Smith and Inohara.

IV. ORDER

In consideration of the foregoing, it is hereby:

ORDERED that claims 2–4, 6, 7, 10, 16–18, 20, 21, 24, and 29–36 of U.S. Patent No. 7,188,145 B2 have not been shown to be unpatentable; and

FURTHER ORDERED that, because this is a Final Written Decision, the parties to the proceeding seeking judicial review of the decision must comply with the notice and service requirement of 37 C.F.R. § 90.2.

PETITIONER:

Eric A. Buresh
Mark C. Lang
ERISE IP, P.A.
eric.buresh@eriseip.com
mark.lang@eriseip.com

PATENT OWNER:

Darren W. Collins
Robert C. Hilton
Aaron J. Pickell
McGUIRE WOODS, LLP
dwcollins@mcguirewoods.com
rhilton@mcguirewoods.com
apickell@mcguirewoods.com
patents@mcguirewoods.com