IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

ORACLE CORPORATION and ORACLE U.S.A. INC.,	)	
Plaintiffs/Counterclaim Defendants,	)	C.A. No. 06-414-SLR
v.	)	**JURY TRIAL DEMANDED**
EPICREALM LICENSING, LP,	)	**PUBLIC VERSION**
Defendant/Counterclaim Plaintiff.	)	

EPICREALM'S MEMORANDUM IN SUPPORT OF ITS MOTION FOR PARTIAL SUMMARY JUDGMENT THAT THE ORACLE REFERENCES DO NOT ANTICIPATE

Richard L. Horwitz (#2246)
David E. Moore (#3983)
POTTER ANDERSON & CORROON LLP
Hercules Plaza, 6th Floor
1313 N. Market Street
Wilmington, DE 19899
Tel: (302) 984-6000
rhorwitz@potteranderson.com
dmoore@potteranderson.com

Attorneys for Defendant/Counterclaim Plaintiff epicRealm Licensing, LP

OF COUNSEL:

Harry J. Roper
George S. Bosy
Aaron A. Barlow
Patrick L. Patras
David R. Bennett
Paul D. Margolis
Benjamin J. Bradford
Emily C. Johnson
JENNER & BLOCK
330 N. Wabash Avenue
Chicago, IL 60611-7603
Tel: (312) 923-8305

Dated: July 31, 2008
876828 / 31393 / Oracle

Public Version Dated: August 7, 2008

TABLE OF CONTENTS

I. NATURE AND STAGE OF THE PROCEEDINGS 1

II. SUMMARY OF ARGUMENT 1

III. CONCISE STATEMENT OF FACTS 2

A. Background Of The Technology 2

B. The '554 And '335 Patents-In-Suit 4

C. Relevant Oracle Products 8

1. Oracle's First Aborted Attempt At Managing Dynamic Web Page Requests: Oracle 1.0 9

2. Oracle's Second Aborted Attempt At Managing Dynamic Web Page Requests: Oracle 2.0 12

3. The Accused Oracle Products 14

4. Oracle Touts The Advantages Of Its Metric-Based Load Balancing In The Accused Oracle Products 15

D. Oracle Files The Instant Litigation 15

IV. ARGUMENT 16

A. Applicable Legal Principles 16

1. Summary Judgment 16

2. Anticipation 17

B. EpicRealm Is Entitled To Partial Summary Judgment That The Oracle References Do Not Anticipate Any Of The Asserted Claims 19

1. The Oracle References Are Not Prior Art 19

a. Oracle Cannot Prove By Clear And Convincing Evidence That Any Of The Oracle References Were Publicly Accessible Or Otherwise Known Or Used By Others. 19

b. Even If Oracle Could Prove Public Accessibility, Knowledge Or Use By Others By Clear And Convincing Evidence, Oracle Still Could Not Prove That Any Of The Oracle References Antedate Epicrealm's Invention. 20

c. Oracle Has No Argument Under 35 U.S.C. Section 102(g). 22

2. The Oracle 1.0 References And The SQL*Net References Do Not Disclose All The Limitations Of Any Of The Asserted Claims. 22

a. Reference 153 Does Not Disclose At Least "Dispatching" Under Either Party's Construction. 22

i. Reference 153 Does Not Disclose "Dispatching" Under Epicrealm's Construction. 23

ii. Reference 153 Does Not Disclose "Dispatching" Under Oracle's Construction. 24

b. Reference 503 Does Not Disclose At Least "Dispatching" Under Either Party's Construction. 25

c. The Oracle 1.0 Product Does Not Disclose At Least "Dispatching" Under Either Party's Construction. 25

d. The PTO Already Determined That The Asserted Claims Were Patentable Over Oracle 1.0. 26

e. The SQL*Net References Do Not Disclose Numerous Limitations, Including At Least "Intercepting," "Page Server," "Concurrent Processing" And "Dispatching" Under Either Party's Construction. 26

i. The SQL*Net 2.2 Reference 27

ii. The SQL*Net 2.3 Reference 29

f. Oracle Cannot Combine References For Anticipation Purposes. 30

g. Even If Combined, The Oracle 1.0 References And The SQL*Net References Do Not Anticipate Any Of The Asserted Claims. 30

3. The Oracle 2.0 References Do Not Anticipate Any Of The Asserted Claims. 31

a. There Is No Evidence That Reference 605 And The Webserver 2.0 Guide Disclose All Of The Limitations Of Any Of The Asserted Claims. 31

b. The Oracle 2.0 References Do Not Disclose "Dispatching" Under Either Party's Construction. 32

i. The Oracle 2.0 References Do Not Disclose "Dispatching" Under EpicRealm's Construction. 33

ii. The Oracle 2.0 References Do Not Disclose "Dispatching" Under Oracle's Construction 34

c. The Oracle 2.0 References Do Not Disclose "Intercepting" Under Either Party's Construction.....35

V. CONCLUSION.....37

TABLE OF AUTHORITIES

Cases

Advanced Display Sys. Inc. v. Kent State Univ.,
212 F.3d 1272 (Fed. Cir. 2000) 30

Am. Hoist & Derrick Co. v. Sowa & Sons,
725 F.2d 1350 (Fed. Cir. 1984) 26

Anderson v. Liberty Lobby, Inc.,
477 U.S. 242 (1986) 17

Beachcombers, Int'l Inc. v. WildeWood Creative Prods., Inc.,
31 F.3d 1154 (Fed. Cir. 1994) 17

Becton Dickinson and Co. v. Tyco Healthcare Group LP,
C.A. No. 04-1694-GMS, 2004 WL 2075413 (D. Del. 2004) 19

Brown v. Barbacid,
436 F.3d 1376 (Fed. Cir. 2006) 22

Burroughs Wellcome Co. v. Barr Labs, Inc.,
40 F.3d 1223 (Fed. Cir. 1994) 20

Connell v. Sears, Roebuck & Co.,
722 F.2d 1542 (Fed. Cir. 1983) 17

Cont'l Can Co. USA v. Monsanto Co.,
948 F.2d 1264 (Fed. Cir. 1991) 17, 34

Eaton v. Evans,
204 F.3d 1094 (Fed. Cir. 2000) 20

In re Hall,
781 F.2d 897 (Fed. Cir. 1986) 18

Hilgraeve, Inc. v. Symantec Corp.,
271 F. Supp. 2d 964 (E.D. Mich. 2003) 19

Horowitz v. Fed. Kemper Life Assurance Co.,
57 F.3d 300 (3d Cir. 1995) 17

Hybritech, Inc. v. Monoclonal Antibodies,
802 F.2d 1367 (Fed. Cir. 1986) 17

Izumi Prods. v. Koninklijke Philips Electronics N.V.,
315 F. Supp. 2d 589 (D. Del. 2004) 19

Mahurkar v. C.R. Bard, Inc.,
79 F.3d 1572 (Fed. Cir. 1996) 20, 22

Mannesmann Demag Corp. v. Engineered Metal Prods. Co.,
226 U.S.P.Q. 466 (D. Del. 1985) 18

Matsushita Elec. Indus. Co. v. Zenith Radio Corp.,
475 U.S. 574 (1986) 16

In re Oelrich,
666 F.2d 578 (C.C.P.A. 1981) 17

Pharmacia & Upjohn Co. v. Sicor Inc.,
447 F. Supp. 2d 363 (D. Del. 2006) 19

Rackable Sys., Inc. v. Super Micro Computer, Inc.,
2007 WL 1223807 (N.D. Cal. 2007) 18

ResQNet.com, Inc. v. Lansa, Inc.,
533 F. Supp. 2d 397 (S.D.N.Y. 2008) 18

Transclean Corp. v. Bridgewood Servs.,
290 F.3d 1364 (Fed. Cir. 2002) 19

Tronzo v. Biomet, Inc.,
156 F.3d 1154 (Fed. Cir. 1998) 17

W.L. Gore & Assocs., Inc. v. Garlock, Inc.,
721 F.2d 1540 (Fed. Cir. 1983) 17

Woodland Trust v. Flowertree Nursery, Inc.,
148 F.3d 1368 (Fed. Cir. 1998) 18

In re Wyer,
655 F.2d 221 (C.C.P.A. 1981) 18

Zenith Elecs. Corp. v. PDI Comm. Sys., Inc.,
522 F.3d 1348 (Fed. Cir. 2008) 18, 30

Statutes

35 U.S.C. § 102 ..... 17, 18, 22

Rules

Fed. R. Civ. P. 56(c) 16

Defendant epicRealm Licensing, LP ("epicRealm") moves this Court for partial summary judgment that certain Oracle references do not anticipate any of the asserted claims of the patents-in-suit.

I. NATURE AND STAGE OF THE PROCEEDINGS

This patent action was filed in 2006 and is now scheduled for a two-week jury trial commencing January 12, 2009. Pursuant to the Stipulated Amendment to Joint Discovery Plan and Scheduling Order (D.I. 191) entered by the Court on July 18, 2008, the deadline for filing summary judgment opening briefs is July 31, 2008.

II. SUMMARY OF ARGUMENT

EpicRealm should be granted partial summary judgment that the Oracle References (as defined herein) do not anticipate any of the asserted claims for two reasons. *First*, Oracle failed to establish that the Oracle References are prior art to the asserted claims. In discovery, Oracle has not provided any evidence that any of the Oracle References (that are documents) were ever publicly accessible, much less publicly accessible prior to the invention date. Similarly, Oracle has not provided any evidence that any of the Oracle References (that are products) were known or used by others in this country prior to the invention date. This alone is a sufficient basis to grant epicRealm's motion. *Second*, none of the Oracle References discloses all of the limitations of any of the asserted claims. For example, none of the Oracle References discloses the "dispatching" limitation recited in all of the asserted claims under either party's proposed construction for "dispatching." Given that the Oracle References are central to Oracle's anticipation defense, epicRealm expects this partial summary judgment motion, if granted, to simplify the trial.

III. CONCISE STATEMENT OF FACTS

This case involves two epicRealm patents: United States Patent Nos. 5,894,554 ("the '554 Patent") (Exh. 1) and 6,415,335 ("the '335 Patent") (Exh. 2.) (See Appendix filed herewith). The '335 Patent is a continuation of the '554 Patent. EpicRealm asserts that Oracle infringes Claims 1-5 and 7-11 of the '554 Patent and Claims 2 and 16 of the '335 Patent (the "asserted claims"). EpicRealm has accused Oracle of infringing the patents-in-suit with respect to the following products: (1) the Oracle Web Cache Products beginning in November 2000 with Release 1.0.2 and all subsequent releases (the "Accused Oracle Web Cache Products"); (2) the Oracle Application Server Products beginning in April 2003 with Release 10gR1 (9.0.4) and all subsequent releases (the "Accused Oracle Application Server Products"); and (3) the Oracle Database Products With Real Application Clusters ("RAC") beginning in May 2005 with Release 10gR2 (10.2.0.1.0) for JDBC and all subsequent releases and beginning in October 2007 with Release 11g (11.1) for OCI and all subsequent releases (the "Accused Oracle Database Products") (all three collectively, the "Accused Oracle Products").

A. Background Of The Technology

The patented invention can best be understood in the context of the prior art and the prior art problems that the patents solved. This prior art included the then existing World Wide Web (the "Web") that was used by a "Web client" (*e.g.*, a computer running a Web browser) desiring to access information on a Web site through the retrieval of Web pages. In this system, the "Web client" used a "Web browser" (*e.g.*, Microsoft Internet Explorer or other software capable of requesting and displaying Web content) to identify the Web site and the requested information; the Web browser connected to the Web, allowing access to the Web site. The Web server operating the Web site received the request and generated a response which was returned to the Web browser. As explained in the patents-in-suit, the prior art used a particular software

language to facilitate this process called "HTML" (HyperText Markup Language). The prior art also used a protocol for certain Web communications called "HTTP" (Hypertext Transport Protocol). (*See* Exh. 1, '554 Patent at col. 1, ll. 24-26.)

Initially, most Web sites provided only "static" Web pages. Typically, static Web pages are files stored on the Web server that remain the same until they are manually modified. (Declaration of Dr. David Finkel ("Finkel Decl.") ¶4 (filed herewith).) When a static Web page is requested, the Web server retrieves the specific file requested by the Web client and returns that file to the Web client without modifying the file. (*Id.*)

As the Web developed, Web sites began to provide dynamic Web pages, *i.e.*, Web pages that are generated only after they are requested in response to the specific request of the Web client. (Finkel Decl. ¶5.) In order to generate dynamic Web pages, the Common Gateway Interface ("CGI") was developed to facilitate the generation of dynamic Web pages by a Web server, as were various "tools" that facilitated the use of CGI. (*See* Exh. 1, '554 Patent at col. 1, ll. 47-63.) The processing of dynamic Web pages is more resource intensive than is the case with static Web pages, *e.g.*, requiring more processor time, memory and/or other resources. (Exh. 3, Shamos Rpt. ¶29.) As the number of dynamic Web page requests increased, many issues arose based on this increased demand for Web server resources, including slower response time, the failure to provide the requested Web content, or even the complete failure ("crashing") of the Web server. (Finkel Decl. ¶5.)

The patents-in-suit summarize some of the problems with the management of dynamic Web page requests by the prior art, including CGI:

> Tools that generate CGI applications do not, however, resolve the problem of managing numerous Web pages and requests at a Web site. For example, a single company may maintain hundreds of Web pages at their Web site. Current Web server architecture also

> does not allow the Web server to efficiently manage the Web page and process Web client requests. Managing these hundreds of Web pages in a coherent manner and processing all requests for access to the Web pages is thus a difficult task. Existing development tools are limited in their capabilities to facilitate dynamic Web page generation, and do not address the issue of managing Web requests or Web sites.

(Exh. 1, '554 Patent at col. 2, ll. 1-12.)

REDACTED

The prior art attempted to solve these problems by adding hardware (*i.e.*, more computers or routers) to a Web site to handle the increase in requests for dynamic Web pages. (Finkel Decl. ¶5.) This approach was costly and cumbersome. (Finkel Decl. ¶5.) As explained below, the patents-in-suit solved the problem by minimizing the need for adding costly and cumbersome hardware.

B. The '554 And '335 Patents-In-Suit

The patents-in-suit generally relate to systems for efficiently managing dynamic Web page generation requests. The patents-in-suit solve the aforementioned problems associated with the management of dynamic Web page requests in a very different way from the prior art attempts to solve those problems (*e.g.*, CGI and related development tools). An example of the

"architecture" of an embodiment of the patented invention is shown in Figure 4 of the patents-in-suit:

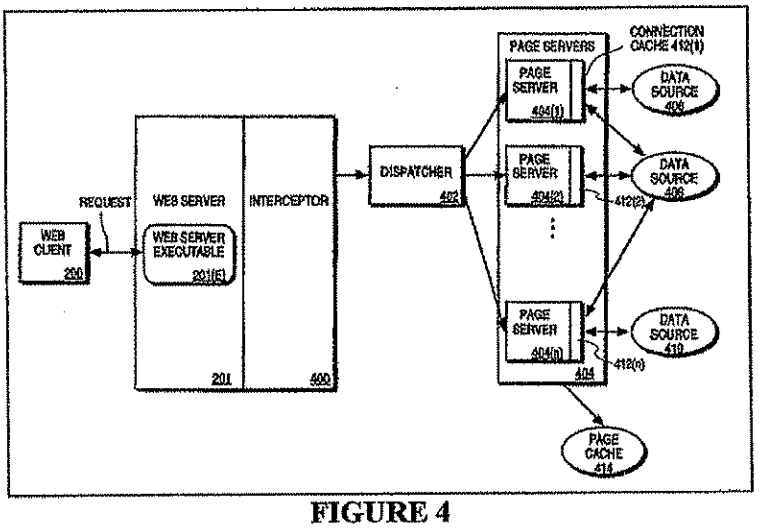


FIGURE 4

(Exh. 1, '554 Patent at Fig. 4.) In the above architecture, starting from the left of the diagram, the Web client (200), which again may simply be a computer running Microsoft Internet Explorer, initiates the request for a static or dynamic Web page, and the request is sent over the Web to the Web server (201) for processing. (*Id.* at col. 4, ll. 55-57.) The Web server may itself process the request and send a Web page back to the Web client. Not all requests are handled by the Web server, according to the patent. For a request that will not be processed at the Web server, "Interceptor 400 intercepts the request and routes it to Dispatcher 402." (*Id.* at col. 4, ll. 58-60; *see also* col. 5, ll. 37-39.) For example, if the request is for a static Web page, *i.e.*, one that is relatively easy to locate and retrieve, the request may be processed by the Web server itself—in other words, the static page is located by the Web server and then sent back to the Web client. If, however, the request is for a dynamic Web page, the Web page to be sent back may be created by one of a number of "page servers" (404) shown to the right in the figure.

In the above illustration, three page servers are shown. However, there could be fewer or more than three page servers depending on the configuration of the Web site. (*See id.* at col. 5,

ll. 37-39.) When the request is received at the Web server, the Web server initially has the duty to process the request. However, when a request is intercepted and routed from the Web server to a page server, that page server assumes the processing duties for that request. In this way, the Web server is "released" to perform other tasks. (*See id.* at col. 5, ll. 8-19, col. 6, ll. 20-24.)

Coming back to Figure 4 above, once a request is intercepted, it is sent to the dispatcher (402). The function of the dispatcher is to select a page server that can more efficiently process the request. First, the dispatcher examines the request to determine the subset of page servers that are capable of processing this type of request. (*See, e.g., id.* at col. 6, ll. 12-13.) Second, the dispatcher makes an informed selection as to which page server in the subset ***can more efficiently process*** the request based on dynamic information maintained about the page servers. (*See, e.g., id.* at col. 6, ll. 14-19.) In that way, the dispatcher can take advantage of the fact that one page server in the subset of capable page servers can more efficiently process the request than the other page servers that can process it. (*Id.* at col. 5, l. 50-col. 6, l. 19.) In this example, the dispatcher greatly increases the efficiency of processing Web page requests. As disclosed in the patents-in-suit, this type of "load balancing" of dynamic Web page requests can significantly increase the performance at a busy Web site. ('554 Patent, col. 6, ll. 16-19.)

As just noted, the selection made by the dispatcher as to which page server can more efficiently process the request is based on dynamic information maintained about page servers. (*Id.* at col. 5, ll. 50-55.) In the context of the patented invention, dynamic information about the page servers has several qualities. It is important to recognize that dynamic information is information that can only be established during the execution of a program. (Finkel Decl. ¶6.) In other words, dynamic information is not information that can be established prior to the execution of the program. (*Id.*) In addition, dynamic information in the context of the patent

must be information maintained about page servers. (Exh. 1, '554 Patent at col. 5, ll. 50-55.) This dynamic information maintained about page servers must indicate which page server can more efficiently process the request. (*Id.* at col. 5, l. 50-col. 6, l. 19.)

Three examples given in the patents-in-suit illustrate the type of dynamic information that allows the dispatcher to determine which page server can more efficiently process the request. As a first example, the dynamic information may identify a page server that has the least processing load (*e.g.*, the number of requests that the page server is servicing) as compared to other page servers, a page server that can process the request more quickly than the other page servers that can process it. (*Id.* at col. 6, ll. 14-19.) This is a type of information that cannot be known until the program is executing because the number of requests being serviced cannot be determined until the page server software is running. The patents-in-suit specifically describe this type of load balancing capability as one that can "significantly increase performance at a busy Web site." (*Id.* at col. 6, ll. 16-19.)

In a second example in the patent, the dynamic information may identify a particular page server that is already logged into a data source, a page server that can access the requested data more quickly than a page server that would need to log into the data source. (*Id.* at col. 6, ll. 5-13.) Again, this is the type of information that cannot be known until the software is running because the page server cannot log into a data source until the page server software is running.

As a final example, the dynamic information may identify a page server that has already retrieved the requested content in a previous request, a page server that therefore does not need to spend time to obtain the requested content from the data source. (*Id.* at col. 6, ll. 1-3.) Yet again, this is the type of information that that cannot be known until the page server software is running.

The summary of the invention set forth in the patents-in-suit explains how the patented invention functions:

> In one embodiment, the present invention claims a computer-implemented method for managing a dynamic Web page generation request to a Web server, the computer-implemented method comprising the steps of routing the request from the Web server to a page server, the page server receiving the request and releasing the Web server to process other requests, processing the request, the processing being performed by the page server concurrently with the Web server, as the Web server processes the other requests, and dynamically generating a Web page in response to the request, the Web page including data dynamically retrieved from one or more data sources.

(Exh. 1, '554 Patent at col. 2, ll. 20-31.)

The patented invention provides many advantages for a Web site host. The patented invention decreases the time necessary to process requests by allowing for dispatching and the subsequent routing to a page server as discussed above. (*Id.* at col. 6, ll. 14-19.) In addition, the patented invention improves performance because it facilitates the scaling up of the Web site to handle more requests as the Web traffic grows. (*Id.* at col. 8, ll. 11-24.) "Scalability" refers to the ability to increase the processing capability of a Web site, *e.g.*, the ability to add page servers as needed. (*Id.*; Finkel Decl. ¶7.)

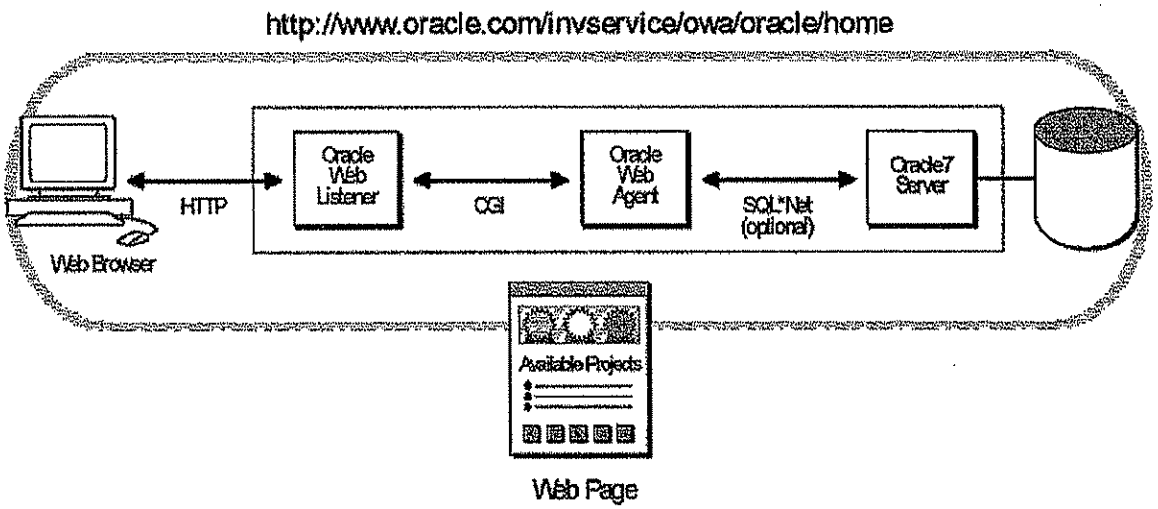
C. Relevant Oracle Products

Over the years, Oracle has attempted to develop many products for generating dynamic Web pages. Oracle aborted many such attempts, prior to invoking the technology of the patents-in-suit. Many years after epicRealm's filing date and after the '554 Patent issued, Oracle adopted what it calls "metric-based load balancing" in the three Accused Oracle Products, which are the subject of epicRealm's Motion for Partial Summary Judgment of Literal Infringement, filed contemporaneously herewith. It is Oracle's earlier, aborted attempts, however, that make up the allegedly anticipatory Oracle References of the instant motion.

1. Oracle's First Aborted Attempt At Managing Dynamic Web Page Requests: Oracle 1.0

Oracle's first attempt to design a system to manage dynamic Web page requests was called Oracle WebServer 1.0 ("Oracle 1.0"). Although Oracle quickly abandoned Oracle 1.0, it now claims that various references relating to Oracle 1.0 anticipate the Asserted Claims. Oracle contends that "Oracle WebServer User's Guide, Release 1.0, Part No. A34986-2" (designated by Oracle as "Reference 153"), "The Oracle WebServer: A Technical Discussion" (designated by Oracle as "Reference 503") and the Oracle 1.0 product (collectively, the "Oracle 1.0 References") anticipate all of the asserted claims. (Exh. 3, Shamos Rpt. ¶¶69, 808.) Oracle also contends that the Oracle 1.0 References can be combined with two other references about a different Oracle product called SQL*Net: "Understanding SQL*Net, Release 2.2" (the "SQL*Net 2.2 Reference") and "Understanding SQL*Net, Release 2.3" (the "SQL*Net 2.3 Reference") (collectively, the "SQL*Net References") and that the combinations anticipate all of the asserted claims. (*Id.* at. ¶110.)

Reference 153 describes and illustrates Oracle 1.0 as shown below:



Web Page

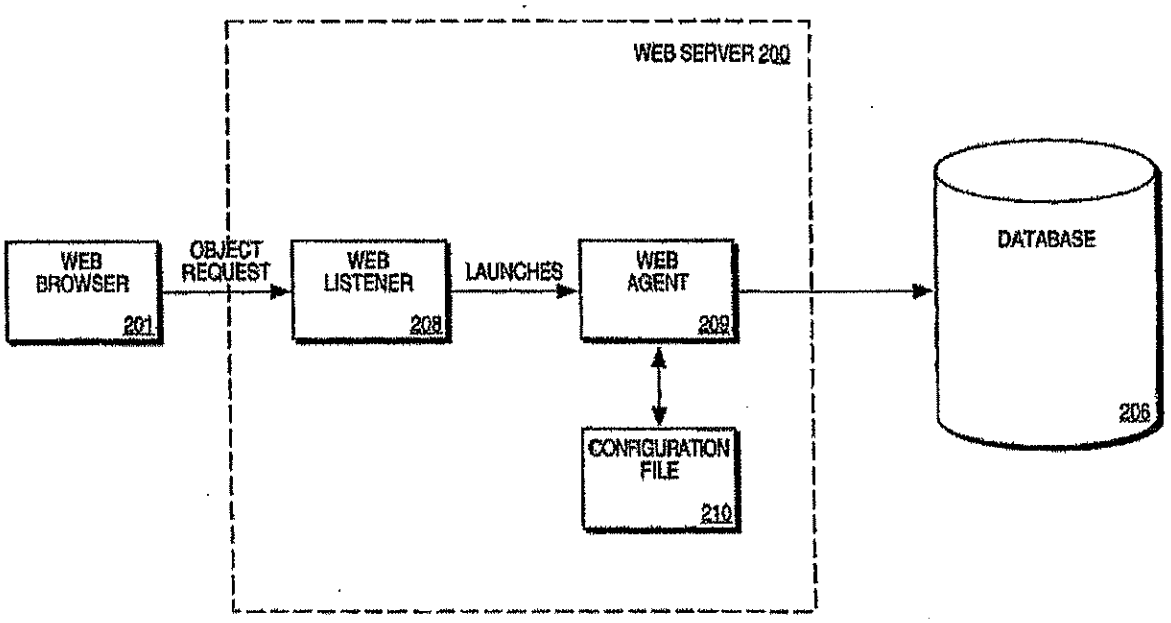
(Exh. 5 at ORCL00941729.) According to Reference 153, the Oracle Web Listener received a request and examined the request to determine whether it was for a dynamic or static page. (*Id.*) For a dynamic Web page request, the Web Listener invoked a new Oracle Web Agent, which is a CGI process, and then the Web Listener sent the request to the newly created Web Agent. (Exh. 5 at ORCL00941822.) The Web Agent parsed the request and accessed a configuration file (not shown). (*Id.*) The configuration file provided information as to what service provided by the Oracle7 Server could process this particular type of dynamic Web page request. (*Id.* at ORCL00941822-23.) The Web Agent then used the information in the request and in the configuration file to access the appropriate application on the Oracle7 Server. (*Id.* at ORCL00941820.) The Oracle7 Server executed the stored PL/SQL procedure that was specified as part of the request. (*Id.* at ORCL00941822.) Importantly, the Oracle Web Agent had no choice as to which Oracle7 Server to connect to or which PL/SQL procedure to execute. (*Id.* at ORCL00941820.) Oracle 1.0 had several performance problems. For example, if many requests required the same application, there was no way to perform any type of load balancing of the requests because there was only a single instance of the application, which had to process all of the requests. This could lead to degraded performance.

REDACTED

REDACTED

The Reference 153 diagram above includes the language "SQL*Net (optional)." SQL*Net was a software utility provided by Oracle for remote data access (the ability to access a database located on another computer). (Exh. 11 at ORCL00106638.) While SQL*Net is discussed in more detail later, it is worth noting SQL*Net could not even generate a dynamic Web page. (Finkel Decl. ¶36.)

Notably, the United States Patent and Trademark Office ("PTO") granted the patents-in-suit over U.S. Patent 5,761,673 (the "Oracle Bookman Patent"), which disclosed the same features relied upon by Oracle in Reference 153 for anticipation purposes. Below is Figure 3 from the Oracle Bookman Patent:

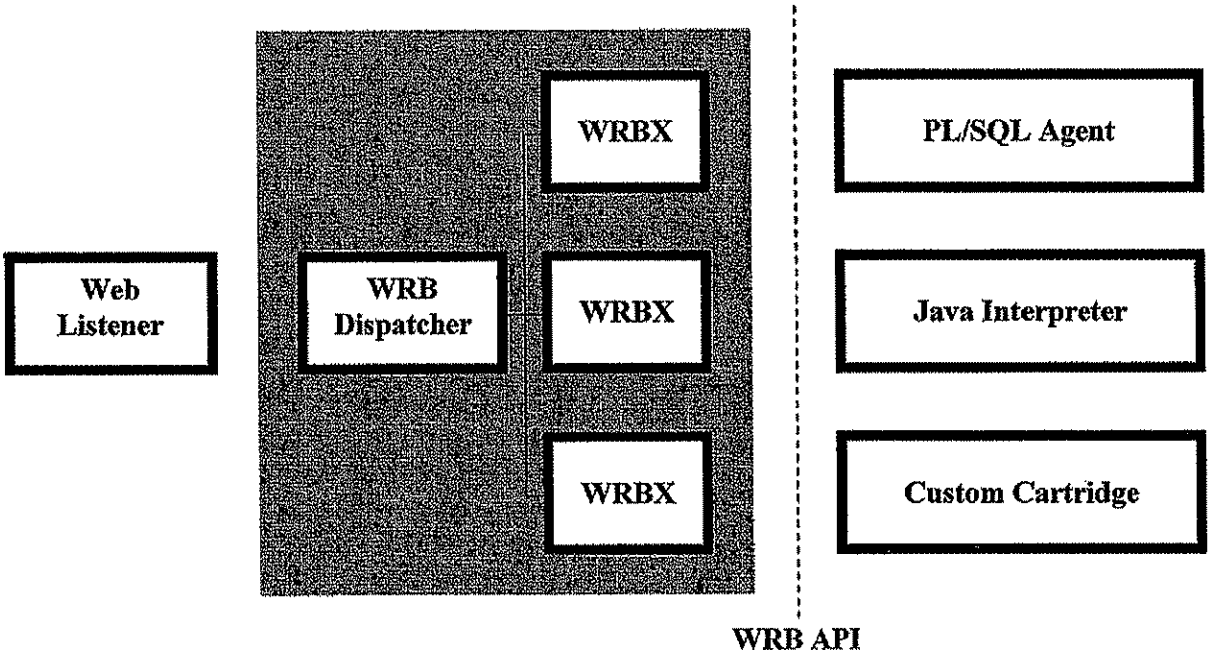


REDACTED

REDACTED Oracle's next attempted solution was Oracle 2.0 (Exh. 13 at ORCL01606516), which did not implement the inventions of the patents-in-suit and was also eventually abandoned by Oracle.

2. Oracle's Second Aborted Attempt At Managing Dynamic Web Page Requests: Oracle 2.0

Oracle next tried to manage dynamic Web page requests in what it called Oracle WebServer 2.0 ("Oracle 2.0"). Although Oracle also abandoned this product, Oracle now contends that various references relating to Oracle 2.0 also anticipate all of the asserted claims. (Exh. 3, Shamos Rpt. ¶¶70, 540, 554.) In particular, Oracle contends that the asserted claims are anticipated by "Oracle WebServer User's Guide, Release 2.0.2 Production, Part No. A23646-2" (designated by Oracle as "References 604"), "Oracle WebServer 2.0, Technical Note, March 1996" (designated by Oracle as "Reference 605", the "Oracle WebServer User's Guide Release 2.0 Production, Part No. A23646-1" (the "WebServer 2.0 Guide") and the Oracle 2.0 product (collectively, the "Oracle 2.0 References"), either individually or in combination. (*Id.*) Below is a rough diagram of Oracle 2.0 drawn from an Oracle document:



(drawn from Exh. 8 at ORCL000766.)

Unlike Oracle 1.0, Oracle 2.0 did not rely solely on Oracle Database to generate dynamic Web pages. In Oracle 2.0, the then-existing Oracle HTTP Server received incoming Web page requests, delivered static files and ran simple CGI programs. (*Id.* at ORCL000764.) Unlike standard Web servers, the HTTP Server did "not make any attempt at interpreting [a request], instead it is immediately handed off to the Web Request Broker for further processing." (*Id.* at ORCL000766.) The Web Request Broker included a Web Request Broker Dispatcher ("WRB Dispatcher") and Web Request Broker Engines ("WRBXs") for generating dynamic Web pages. (*Id.*)

The job of the WRB Dispatcher was to route requests to WRBXs which processed the requests, *e.g.*, generated dynamic Web pages. (*Id.*) When routing a request, the WRB Dispatcher first examined a configuration file to determine which WRBXs were configured to process the request. (*Id.*) Next, the WRB Dispatcher checked to see which of the properly configured WRBXs were not already processing a request and, thus, could process the current

request. (Exh. 9 at ORCL000804.) The WRB Dispatcher then sent the request to one such WRBX.[1] (Exh. 8 at ORCL000766.) If there were none, the WRB Dispatcher could start a new WRBX process unless a predetermined maximum number of WRBX processes had been reached. (*Id.*) Finally, if the WRB Dispatcher could not find a WRBX that was properly configured to process a request, the request was passed back to the HTTP Server for standard processing (*e.g.*, static Web page retrieval and error handling). (*Id.*)

Thus, the WRB Dispatcher was incapable of using any information about the relative efficiencies of the WRBXs that were properly configured to process the request and are not already processing a request. This is not surprising because the processing efficiencies of such WRBXs were identical.

Oracle's aborted Oracle 1.0 and Oracle 2.0 products form a central portion of Oracle's anticipation defense. The instant motion seeks partial summary judgment that neither the Oracle 1.0 References (with or without the SQL*Net References discussed below) nor the Oracle 2.0 References anticipate any of the asserted claims.

3. The Accused Oracle Products

Years after Oracle 1.0 and Oracle 2.0, and after epicRealm's '554 Patent issued, Oracle began using what it calls "metric-based load balancing" of dynamic Web page requests in its products. This "metric-based load balancing" functionality, which was not present in Oracle 1.0 or Oracle 2.0, corresponds to the "dispatching" limitation in the asserted claims. Thus, the Oracle products epicRealm has accused of infringing include functionality that was not part of Oracle 1.0 or Oracle 2.0. This is not a case where Oracle is relying for anticipation on the same

[1] The Oracle 2.0 documentation is silent as to how the WRB Dispatcher chose between such WRBXs.

technology that is accused of infringement. Oracle's own actions illustrate its sea change in technology.

4. Oracle Touts The Advantages Of Its Metric-Based Load Balancing In The Accused Oracle Products.

Oracle holds an annual convention for its customers and potential customers called Oracle Open World. (Exh. 28, Surber Dep. 99:2-18.)

REDACTED (*Id.*) At Oracle Open World, Oracle made a presentation on metric-based load balancing telling its customers and potential customers that using metric-based load balancing was "the best way not to fall," *i.e.*, the best way to prevent a Web site from crashing. (Exh. 14 at ORCL01050248-64; Exh. 28, Surber Dep. 98:16-103:10.) The presentation highlighted the ability of the Accused Oracle Web Cache Products, the Accused Oracle Application Server Products and the Accused Oracle Database Products to use metric-based load balancing, even stating that metric-based load balancing used in the Accused Oracle Application Server Products and in the Accused Oracle Database Products had the highest possible level of criticality.

D. Oracle Files The Instant Litigation

REDACTED

Oracle contends that the asserted claims are anticipated by a collection of 30 references, including the Oracle 1.0 References (with or without the SQL*Net References) and the Oracle 2.0 References (collectively, the "Oracle References"). (Exh. 3, Shamos Rpt. ¶¶195, 808, 809.)

IV. ARGUMENT

EpicRealm should be granted partial summary judgment that the Oracle References do not anticipate any of the asserted claims because the Oracle References are not prior art and none of the Oracle References discloses all of the limitations of any asserted claim under either party's constructions.

More specifically, Oracle cannot prove by clear and convincing evidence that any of the Oracle References (that are documents) were ever publicly available or accessible. Similarly, Oracle cannot prove by clear and convincing evidence that any of the Oracle References (that are products) constitute products that were known or used by others prior to the invention date.

Even if Oracle could prove that the Oracle References are prior art, Oracle's anticipation argument would still be fatally flawed. None of the Oracle References discloses the "dispatching" limitation that is recited in every asserted claim. Oracle did not introduce this functionality until after the '554 Patent issued when Oracle added the metric-based load balancing feature into its products. Oracle 1.0, Oracle 2.0 and the Oracle References simply lack this feature. For these reasons, the Court should grant epicRealm's motion for partial summary judgment that the Oracle References do not anticipate any of the asserted claims.

A. Applicable Legal Principles

The law on summary judgment and anticipation is well settled.

1. Summary Judgment

Summary judgment should be granted if the "pleadings, the discovery and disclosure materials on file, and any affidavits show that there is no genuine issue as to any material fact and that the movant is entitled to judgment as a matter of law." Fed. R. Civ. P. 56(c). The moving party bears the burden to prove that there are no genuine issues of material fact. *Matsushita Elec. Indus. Co. v. Zenith Radio Corp.*, 475 U.S. 574, 586 n.10 (1986). A fact that

can "alter the outcome [is] material." *Horowitz v. Fed. Kemper Life Assurance Co.*, 57 F.3d 300, 302 n.1 (3d Cir. 1995). An issue as to a fact is "genuine...if the evidence is such that a reasonable jury could return a verdict for the nonmoving party." *Anderson v. Liberty Lobby, Inc.*, 477 U.S. 242, 248 (1986).

2. Anticipation

Anticipation under 35 U.S.C. Section 102 requires that each and every element of the claimed invention be identically disclosed in a single prior art reference. *Hybritech, Inc. v. Monoclonal Antibodies*, 802 F.2d 1367, 1379 (Fed. Cir. 1986); *W.L. Gore & Assocs., Inc. v. Garlock, Inc.*, 721 F.2d 1540, 1554 (Fed. Cir. 1983). Anticipation also requires that the elements in the prior art reference be arranged in the same manner as in the claim. *Connell v. Sears, Roebuck & Co.*, 722 F.2d 1542, 1548 (Fed. Cir. 1983).

An element may be inherently disclosed in a prior art reference. *See In re Oelrich*, 666 F.2d 578, 581 (C.C.P.A. 1981). In order for a disclosure to be inherent, "the missing descriptive matter must necessarily be present...such that one skilled in the art would recognize such a disclosure." *Tronzo v. Biomet, Inc.*, 156 F.3d 1154, 1159 (Fed. Cir. 1998). Inherency may not be established by "probabilities or possibilities." *Cont'l Can Co. USA v. Monsanto Co.*, 948 F.2d 1264, 1268-69 (Fed. Cir. 1991). "The mere fact that a certain thing *may* result from a given set of circumstances is not sufficient" to establish anticipation. *Id.* at 1269 (emphasis in original, citations omitted). In deciding the issue of anticipation, the Court follows a two-step analysis: first, the asserted claims must be construed in order to determine their scope and meaning, and second, each construed claim limitation must be compared to the prior art reference to determine whether the reference discloses that limitation. *Beachcombers, Int'l Inc. v. WildeWood Creative Prods., Inc.*, 31 F.3d 1154, 1160 (Fed. Cir. 1994).

Oracle appears to argue that each of the Oracle References is either a "printed publication" under Section 102(a), or a product allegedly establishing that the claimed invention was "known or used by others in this country" under Section 102(a). Public accessibility is the "touchstone" as to whether a document qualifies as a "printed publication." *In re Hall*, 781 F.2d 897, 899 (Fed. Cir. 1986). Under Section 102(a), such public dissemination or accessibility of the prior art must occur prior to the invention date. *Id.*; *Rackable Sys., Inc. v. Super Micro Computer, Inc.*, 2007 WL 1223807, at *23 (N.D. Cal. 2007). A party asserting that a printed publication is prior art must adduce "sufficient proof of its dissemination or that it has otherwise been available and accessible to persons concerned with the art to which the document relates and thus most likely to avail themselves of its contents." *In re Wyer*, 655 F.2d 221, 227 (C.C.P.A. 1981); *see also Mannesmann Demag Corp. v. Engineered Metal Prods. Co.*, 226 U.S.P.Q. 466, 469-70 (D. Del. 1985).

Under Section 102(a), a person is entitled to a patent unless "the invention was known or used by others in this country ... before the invention thereof by the applicant for patent." Although the case law has not assigned a precise meaning to "known or used by others," it is clear that the knowledge or use must be "available to the public" and proven by clear and convincing evidence. *Woodland Trust v. Flowertree Nursery, Inc.*, 148 F.3d 1368, 1370 (Fed. Cir. 1998).

The moving party is entitled to summary judgment where the nonmoving party fails to adduce enough evidence from which a reasonable jury could conclude by clear and convincing evidence that the invention was known or used by others in this country or was disclosed in a publicly available printed publication prior to the invention date. *Zenith Elecs. Corp. v. PDI Comm. Sys., Inc.*, 522 F.3d 1348, 1356-57 (Fed. Cir. 2008); *ResQNet.com, Inc. v. Lansa, Inc.*,

533 F. Supp. 2d 397, 414 (S.D.N.Y. 2008); *Hilgraeve, Inc. v. Symantec Corp.*, 271 F. Supp. 2d 964, 975 (E.D. Mich. 2003). In addition, summary judgment of no anticipation is appropriate where no genuine issue of material fact exists as to whether a reference is missing a limitation of the claimed invention. *See, e.g., Transclean Corp. v. Bridgewood Servs.*, 290 F.3d 1364, 1372 (Fed. Cir. 2002); *Pharmacia & Upjohn Co. v. Sicor Inc.*, 447 F. Supp. 2d 363, 377 (D. Del. 2006); *Izumi Prods. v. Koninklijke Philips Electronics N.V.*, 315 F. Supp. 2d 589, 603-04 (D. Del. 2004); *Becton Dickinson and Co. v. Tyco Healthcare Group LP*, C.A. No. 02-1694-GMS, 2004 WL 2075413, at *7-8 (D. Del. 2004).

B. EpicRealm Is Entitled To Partial Summary Judgment That The Oracle References Do Not Anticipate Any Of The Asserted Claims.

EpicRealm is entitled to partial summary judgment that the Oracle References do not anticipate any of the asserted claims because the Oracle References are not prior art and do not disclose at least the "dispatching" limitation that is recited in all of the asserted claims.

1. The Oracle References Are Not Prior Art.

The Oracle References are not prior art for two reasons: (1) Oracle cannot prove by clear and convincing evidence that they were publicly accessible or otherwise known or used by others; and (2) even if Oracle could prove that, Oracle cannot prove by clear and convincing evidence that the Oracle References antedate epicRealm's invention date.

a. Oracle Cannot Prove By Clear And Convincing Evidence That Any Of The Oracle References Were Publicly Accessible Or Otherwise Known Or Used By Others.

Oracle cannot prove by clear and convincing evidence that any of the Oracle References that are documents were ever publicly accessible. Similarly, Oracle cannot prove by clear and convincing evidence that any of the Oracle References that are products were ever known or

used by others in this country. As such, Oracle cannot meet its burden of proving by clear and convincing evidence that any of the Oracle References could possibly qualify as prior art.

b. Even If Oracle Could Prove Public Accessibility, Knowledge Or Use By Others By Clear And Convincing Evidence, Oracle Still Could Not Prove That Any Of The Oracle References Antedate Epicrealm's Invention.

Oracle's burden goes beyond merely antedating the filing date of the patents-in-suit. This is so because epicRealm has submitted incontrovertible evidence of an invention date earlier than the April 23, 1996 filing date. The law on proving an invention date earlier than the filing date is well settled. A patent owner can prove an invention date by proving an actual reduction to practice. *Mahurkar v. C.R. Bard, Inc.*, 79 F.3d 1572, 1577 (Fed. Cir. 1996). An actual reduction to practice occurs when a person constructed an embodiment that meets every element of the claimed invention, and the embodiment operated for its intended purpose. *Eaton v. Evans*, 204 F.3d 1094, 1097 (Fed. Cir. 2000). For an actual reduction to practice, the invention must have been sufficiently tested to demonstrate that it will work for its intended purpose, but it need not be in a commercially satisfactory stage of development. *Mahurkar*, 79 F.3d at 1577. Conception occurs when "the inventor had an idea that was definite and permanent enough that one skilled in the art could understand the invention" as it is thereafter to be applied in practice. *Burroughs Wellcome Co. v. Barr Labs, Inc.*, 40 F.3d 1223, 1228 (Fed. Cir. 1994). In order to rely on the conception date as the invention date, the inventor must demonstrate "reasonable diligence" in reducing the invention to practice during the time between conception and reduction to practice. *Mahurkar*, 79 F.3d at 1578.

REDACTED

REDACTED

c. Oracle Has No Argument Under 35 U.S.C. Section 102(g).

In its 351-page, Sixth Supplemental Response To epicRealm's Interrogatory No. 3 on invalidity, Oracle twice mentions 35 U.S.C. § 102(g). Oracle's expert on invalidity issues, Dr. Shamos, did not mention Section 102(g) in his report. Thus, it appears that Oracle has abandoned this defense. If it has not, epicRealm should be granted partial summary judgment on this issue insofar as it relates to the Oracle References.

To prevail on a Section 102(g) defense, Oracle must prove by clear and convincing evidence that another conceived of the claimed invention in this country before the invention by epicRealm and diligently reduced it to practice. *See Brown v. Barbacid*, 436 F.3d 1376, 1379 (Fed. Cir. 2006); *Mahurkar*, 79 F.3d at 1578. Oracle must also show that the alleged prior invention was not abandoned, suppressed or concealed. *Brown*, 436 F.3d at 1379. Oracle cannot meet its burden of proof on any of these issues and has not done so.

2. The Oracle 1.0 References And The SQL*Net References Do Not Disclose All The Limitations Of Any Of The Asserted Claims.

Even if Oracle proved by clear and convincing evidence that the Oracle 1.0 References and the SQL*Net References were prior art, Oracle cannot prove by clear and convincing evidence that they anticipate any of the asserted claims because, at a minimum, they do not disclose the "dispatching" limitation recited in all of the asserted claims.

a. Reference 153 Does Not Disclose At Least "Dispatching" Under Either Party's Construction.

Reference 153 fails to anticipate the asserted claims because, at a minimum, it does not disclose the "dispatching" limitation under either party's construction.[2] Although Oracle's expert, Dr. Shamos, failed to indicate what claim construction, if any, he used for his analysis,

[2] The parties' proposed constructions are referenced throughout and also submitted in the supporting appendix as Exhibit 10.

Dr. Shamos contended that Reference 153 disclosed "dispatching" by what he terms "configuration file dispatching"—a term he apparently made up. (Exh. 3, Shamos Rpt. ¶63.) According to Oracle, Dr. Shamos's "configuration file dispatching" occurs when the Web Agent determines which service in the Oracle7 Server to use based on information contained in the dynamic Web page request and the configuration file. (*Id.* at ¶106; Exh. 4, Shamos Rpt. Ex. 3 at 38.) As shown below, Dr. Shamos's "configuration file dispatching" is not "dispatching" as that term is used in the asserted claims under either party's construction.

i. Reference 153 Does Not Disclose "Dispatching" Under Epicrealm's Construction.

EpicRealm's construction for "dispatching" is: "examining a request to make an informed selection of which page server should process the request based on dynamic information maintained about page servers, the dynamic information indicating which page server can more efficiently process the request, and sending the request to the selected page server." Oracle's argument that Dr. Shamos's "configuration file dispatching" meets the "dispatching" limitation of the claims under epicRealm's construction fails for at least three reasons.

First, the configuration file does not provide any information as to which page server can more efficiently process the request. (Finkel Decl. ¶37.) Instead, the configuration file provides information about the *only* application on an Oracle7 server that can process a specific request. (Exh. 3, Shamos Rpt. ¶ 548; Exh. 5 at ORCL00941822-26.) The configuration file does not provide any comparative information on which page server can more efficiently process a request. (*Id.*)

Second, the information in the configuration file is not "dynamic" information. Dynamic information is information that can only be determined during the execution of a program.

(Finkel Decl. ¶6.)

REDACTED

If Dr. Shamos's "configuration file dispatching" theory were correct, all eight mod_oc4j routing algorithms in the Accused Oracle Application Server Products would meet the "dispatching" limitation.

REDACTED

Third, the Oracle Web Agent does not make an "informed ***selection***."

REDACTED

Oracle contends that an application on an Oracle7 Server is the "page server" (Exh. 3, Shamos Rpt. ¶ 548) and Reference 153 discloses that there is only one that can process the request. (*See* Exh. 5 at ORCL00941822.) Thus, the Oracle Web Agent does not choose from more than one "page server" that can process a request. For all these reasons, Reference 153 fails to disclose "dispatching" under epicRealm's construction.

ii. Reference 153 Does Not Disclose "Dispatching" Under Oracle's Construction.

Oracle's construction for "dispatching" is "analyzing a request to make an informed selection of which page server should process the request, and sending the request to that page server." Reference 153 does not disclose "dispatching" under Oracle's construction for three reasons.

REDACTED

REDACTED

Thus, the Web Agent did not make an "informed selection," because, as also explained above, the Web Agent did not use "dynamic information about the page servers."

Third, the Oracle Web Agent also does not make an "informed selection" because it does not use any information "to make a reasonable or accurate determination as to which [page server] could more efficiently process the request." Thus, Reference 153 does not even measure up to Oracle's own standard of what would constitute "dispatching."

For these reasons, Reference 153 cannot anticipate any of the asserted claims because, at a minimum, it fails to disclose the "dispatching" limitation recited in all of the asserted claims under either party's construction.

b. Reference 503 Does Not Disclose At Least "Dispatching" Under Either Party's Construction.

Reference 503 does not disclose "dispatching" for at least the same reasons set forth above for Reference 153. Reference 503 discloses many of the same concepts as Reference 153—the same concepts considered by the PTO when it granted the patents-in-suit over the Oracle Bookman Patent. (Exh. 6 at ORCL01042971.) Thus, Reference 503 fails to disclose "dispatching" under either party's construction for all the reasons set forth for Reference 153.

c. The Oracle 1.0 Product Does Not Disclose At Least "Dispatching" Under Either Party's Construction.

The Oracle 1.0 product cannot anticipate any of the asserted claims because the only evidence relied upon by Oracle on the operation of the Oracle 1.0 product is References 153 and

503, neither of which discloses, at least, the "dispatching" limitation of the asserted claims under either party's construction.

d. The PTO Already Determined That The Asserted Claims Were Patentable Over Oracle 1.0.

As mentioned above, the PTO granted the patents-in-suit over the Oracle Bookman Patent, which is listed on the cover of both patents-in-suit in the "References Cited" section. (*See* coverpages of Exh. 1, the '554 Patent and Exh. 2, the '335 Patent.) "When an attacker simply goes over the same ground travelled by the PTO, part of the *burden* is to show that the PTO was wrong in its decision to grant the patent." *Am. Hoist & Derrick Co. v. Sowa & Sons*, 725 F.2d 1350, 1360 (Fed. Cir. 1984) (emphasis in original). Because the Oracle 1.0 References are merely cumulative of the Oracle Bookman Patent considered by the PTO, Oracle has the added burden to demonstrate that the PTO was wrong in its decision to grant the asserted claims over the Oracle Bookman Patent. (*Id.*)

e. The SQL*Net References Do Not Disclose Numerous Limitations, Including At Least "Intercepting," "Page Server," "Concurrent Processing" And "Dispatching" Under Either Party's Construction.

Oracle does not argue that either of the SQL*Net References anticipates any of the asserted claims. Nor could it as the SQL*Net References fail to disclose many limitations including "intercepting," "page server," "concurrent processing," and "dispatching." (Finkel Decl. ¶38.) Rather, Oracle argues that the SQL*Net References disclose "dispatching" through the disclosure of "SQL*Net dispatching"—another term apparently coined by Oracle's expert, Dr. Shamos, for purposes of this litigation. Like his other fanciful term, "configuration file dispatching," "SQL*Net dispatching" is not "dispatching" under either party's construction, notwithstanding Dr. Shamos's naming conventions.

i. The SQL*Net 2.2 Reference

Oracle contends that the SQL*Net 2.2 Reference discloses "dispatching" based upon how SQL*Net 2.2 handles a "Multi-Threaded Server" Connection Request. (Exh. 3, Shamos Rpt. ¶110.) A multi-threaded server allows multiple clients (for example, applications requesting data from a database) to connect to the same database server without having a separate, dedicated server process for each client. (Exh. 19 at ORCL01806667.) When a client wishes to connect to a multi-threaded database using SQL*Net 2.2, the client sends to the database "listener" a request for a database connection. (*Id.* at ORCL018066678.) The "listener" receives the request for a database connection and checks whether the client has proper authorization to access the database. (*Id.*) The "listener" then issues a redirect message back over the Web to the client with the address of the "least-called" database server. (*Id.*) The redirect message instructs the client to close its connection with the "listener" and to establish a new, second connection, now with the database whose address was provided in the redirect message, by sending that database a new request for a connection, again over the Web. (*Id.*) Oracle refers to this type of load balancing, *i.e.*, the load balancing that occurs "when a physical connection is first created by the application" to a database, as "Connection Load Balancing." (*See, e.g.*, Exh. 20 at ORCL01060153.)

Establishing a connection to a multi-threaded server in SQL*Net 2.2 does not meet the "dispatching" limitation of any of the asserted claims for at least three reasons.

First, a request for a database connection is not the same as a dynamic Web page request as recited in all of the asserted claims, which is a type of "work request." A request for a database connection attempts to establish a connection between a client and a server; under no circumstances is the response to a request for a database connection going to be a dynamic Web page. (Finkel Decl. ¶39.) By comparison, a dynamic Web page request typically results in a

dynamic Web page in response and cannot establish a connection between a client and a database. (*Id.*)

Second, the SQL*Net 2.2 Reference does not disclose "dispatching" under either party's construction because it does not disclose making an "informed selection." The SQL*Net 2.2 Reference only discloses that the client is redirected to connect to the address of the "least-called" database. Determining the "least called" database can be accomplished, for example, via a round-robin algorithm. (Finkel Decl. ¶40.) Oracle contends that a round-robin algorithm does not meet either party's construction of "dispatching." (Exh. 23, Clark Dep. 103:7-19.) Because the SQL*Net 2.2 Reference does not expressly disclose making an "informed selection," a requirement for both parties' construction of the "dispatching" limitation, the SQL*Net 2.2 Reference cannot anticipate any of the asserted claims.

Third, the SQL*Net 2.2 Reference does not disclose "dispatching" because SQL*Net 2.2 uses a "redirect message" and, thus, does not dispatch a request. Both parties' constructions for "dispatching" require, "sending the request to the selected page server."[3] Thus, even assuming that: (1) a request for a database connection is the same as a dynamic Web page request; and (2) SQL*Net 2.2 makes an informed selection of a database, SQL*Net 2.2 still does not perform "dispatching" because it does not send the request to the selected Oracle7 database as required by both parties' constructions for two reasons. SQL*Net 2.2 does not send the request to the database, instead it sends a redirect message to the client which then, in turn, sends a request for a database connection to the database to establish a connection. (Exh. 19 at ORCL01806668-69.) Hence, SQL*Net is not performing "dispatching" because the client, not SQL*Net 2.2, sends the request for a connection to the database. In addition, the request for a database

[3] EpicRealm's construction includes: "...and sending the request to the selected page server." Oracle's construction includes: "... and sending the request to that page server."

connection sent from the client to the database is a wholly different request for a connection than the request originally received by SQL*Net. (Finkel Decl. ¶41.) After the client receives the redirect message, it closes the connection with the "listener" and issues a new request over the Web to establish a connection with the database. (Exh. 19 at ORCL01806668.) Therefore, the use of a "redirect message" by SQL*Net 2.2 prevents it from "dispatching" under either party's construction.

ii. The SQL*Net 2.3 Reference

Although SQL*Net 2.3 provides additional functionality over SQL*Net 2.2, that additional functionality is irrelevant to the "dispatching" limitation of the asserted claims. The primary difference between SQL*Net 2.3 and SQL*Net 2.2 is that SQL*Net 2.3 discloses what Oracle calls "listener load balancing." (Exh. 21 at ORCL01808646-48.) Under "listener load balancing," requests for database connections sent by clients are randomly distributed among multiple listeners. (*Id.*) After a listener receives a request for a connection in SQL*Net 2.3, it behaves identically as it did in SQL*Net 2.2 (using a redirect message to direct a client to connect to the least called dispatcher). (*Id.*)

The passage from the SQL*Net 2.3 Reference relied upon by Oracle for its "dispatching" analysis does not disclose any additional relevant information to "dispatching" not disclosed by the SQL*Net 2.2 Reference. (Exh. 3, Shamos Rpt. ¶111.) The establishment of a connection by the listener in SQL*Net 2.3 is not "dispatching" under either party's construction for the three reasons set forth above for SQL*Net 2.2. In addition, the random selection of a listener is not "dispatching" under either party's construction because: (1) Oracle concedes that a random selection is not an "informed selection;" (Exh. 27, Shamos Dep. 171:19-172:17) and (2) the listener is not a "page server" under either party's construction because it cannot generate a dynamic Web page.

f. Oracle Cannot Combine References For Anticipation Purposes.

With respect to References 153 and 503, Oracle argues that "[t]hese documents can be considered a single publication because both emanate from the same manufacturer, Oracle Corporation and both describe the same product." (Exh. 3, Shamos Rpt. ¶219.) Similarly, Oracle apparently argues that because "Reference 153 makes the express statement that SQL*Net may optionally be used between the Web Agent and the Oracle7 Servers," Oracle can combine for anticipation purposes various Oracle 1.0 References and various SQL*Net References.

Oracle is wrong as a matter of law. Anticipation of a claim requires that all of the claim elements be found within the four corners of a single reference. *Advanced Display Sys. Inc. v. Kent State Univ.*, 212 F.3d 1272, 1282 (Fed. Cir. 2000). Material not expressly or inherently contained in a single reference may be considered for anticipation purposes only if the material is properly incorporated by reference with sufficient specificity. *Id.* Further, anticipation by the combination of products requires clear and convincing evidence that the products were publicly available and used together in an anticipatory manner. *Zenith, Inc.*, 522 F.3d at 1356-57. That is not the case with any of the Oracle References, including the products and Oracle has never argued that it is. Therefore, for anticipation purposes, each Oracle Reference must be analyzed individually.

g. Even If Combined, The Oracle 1.0 References And The SQL*Net References Do Not Anticipate Any Of The Asserted Claims.

Finally, even if Oracle could show that it is entitled to combine the Oracle 1.0 References and the SQL*Net References, the combination would not anticipate any of the asserted claims for two reasons. *First,* Oracle failed to demonstrate how each and every element of the asserted

claims is disclosed by the combination of the Oracle 1.0 References and the SQL*Net References. For example, Oracle failed to demonstrate that the combination of the Oracle 1.0 References and the SQL*Net References discloses a "Web server," a "page server," "intercepting," and "releasing." *Second*, there is no disclosure of the "dispatching" limitation in any of the Oracle 1.0 References and/or the SQL*Net References under either party's construction for the reasons explained above. For these reasons, the Oracle 1.0 References and the SQL*Net References do not anticipate any of the asserted claims.

3. The Oracle 2.0 References Do Not Anticipate Any Of The Asserted Claims.

The Oracle 2.0 References, individually or collectively, also do not anticipate any of the asserted claims. Oracle failed to present an anticipation argument based on any single Oracle 2.0 Reference. Indeed, Oracle failed to offer any evidence that Reference 605 and the WebServer 2.0 Guide disclose every element of the asserted claims. Therefore, Oracle attempted to improperly combine several Oracle 2.0 References to cobble together anticipation arguments based on Oracle 2.0. Even if it were proper to combine the Oracle 2.0 References for anticipation purposes, the Oracle 2.0 References collectively do not anticipate the asserted claims because they do not disclose at least the "dispatching" and "intercepting" limitations under either party's constructions.

a. There Is No Evidence That Reference 605 And The Webserver 2.0 Guide Disclose All Of The Limitations Of Any Of The Asserted Claims.

Reference 605 and the WebServer 2.0 Guide cannot anticipate any of the asserted claims because Oracle failed to offer any evidence that either reference discloses all of the limitations of any of the asserted claims. Oracle fails to offer any evidence that Reference 605 and the WebServer 2.0 Guide disclose at least the following elements of the asserted claims:

'554 Patent

Claim 1	Claim 9	Claim 11
said page server receiving said request and	wherein said routing further includes intercepting said request at said second computer	said page server receiving said request and
releasing said Web server to process other requests	said page server receiving said request and	releasing said Web server to process other requests
wherein said routing step further includes the steps of intercepting said request at said Web server	releasing said second computer system to process other requests	intercepting said request at said Web server

'335 Patent

Claim 2	Claim 16
said page server receiving said request and	said page server receiving said request and
releasing said Web server to process other requests	releasing said HTTP-compliant device to process other requests
wherein said routing step further includes the steps of intercepting said request at said Web server	wherein said transferring step further includes the steps of intercepting said request at said HTTP-compliant device and

(Exh. 4, Shamos Rpt. Ex. 3 at 157-68.) Therefore, partial summary judgment of no anticipation by Reference 605 and the WebServer 2.0 Guide is proper because Oracle has failed to offer any evidence that those references disclose all of the limitations of any of the asserted claims.

b. The Oracle 2.0 References Do Not Disclose "Dispatching" Under Either Party's Construction.

The Oracle 2.0 References do not disclose the "dispatching" limitation of the asserted claims under either party's construction. Oracle contends that the Oracle 2.0 References disclose "dispatching" when the Oracle Web Request Dispatcher ("WRB Dispatcher") determines which Web Request Broker Engine "(WRBX)" should process a request. (Exh. 4, Shamos Rpt. Ex. 3 at 158.) Specifically, Oracle contends that "[t]he WRB Dispatcher makes an informed selection of which page server (WRBX) should process the request based on dynamic information: which WRBX is free and which one is configured to run the desired service." (*Id.*) Oracle also relied on the disclosure that the WRB Dispatcher can create and destroy WRBXs according to

workload. (*Id.*) Finally, Oracle relied on the disclosure from Reference 605 that "[t]he WRB Dispatcher performs dynamic load balancing between multiple instances of a WRB Service, and the Webmaster can configure each WRB Service to have a minimum and maximum number of instances." (*Id.* quoting Exh. 8 at ORCL000766.) As explained below, this Oracle 2.0 functionality fails to disclose "dispatching" under either party's construction.

i. The Oracle 2.0 References Do Not Disclose "Dispatching" Under EpicRealm's Construction.

EpicRealm's construction for "dispatching" is: "examining a request to make an informed selection of which page server should process the request based on dynamic information maintained about page servers, the dynamic information indicating which page server can more efficiently process the request, and sending the request to the selected page server." The Oracle 2.0 References do not disclose "dispatching" under epicRealm's construction for two reasons.

First, the Oracle 2.0 References fail to disclose maintaining dynamic information about page servers that indicates which page server can more efficiently process the request. As an initial matter, information in the configuration file (*i.e.*, which WRBX is configured to service a particular type of request) is not "dynamic" information as already explained with respect to the Oracle 1.0 References.[4] Moreover, information about which WRBX is configured to service a particular type of request and whether a WRBX is available to process a request does not indicate which WRBX (page server) ***can more efficiently process*** a request. Instead, that information merely indicates which WRBXs (page servers) are available to process a request.

[4] Again, if Oracle really believed that using the information in a configurations file could satisfy the "dispatching" limitation, it would concede that, for example, the Accused Oracle Application Server Products, which use information in a configuration file, meet the "dispatching" limitation. Instead, Oracle contends that such products do not perform "dispatching" under either party's construction. (Exh. 23, Clark Dep. 104:7-19; 127:4-128:6.)

There is no disclosure in the Oracle 2.0 References that the WRB Dispatcher selects which WRBX should process the request based on dynamic information indicating which page server can more efficiently process the request. Thus, the Oracle 2.0 References fail to disclose "dispatching" under epicRealm's construction because they do not disclose maintaining dynamic information about page servers that indicates which page server can more efficiently process the request.

Second, the Oracle 2.0 References fail to disclose making an "informed selection" of which page server should process a request. Anticipation cannot be shown by mere "probabilities or possibilities" nor is it sufficient that a certain thing "*may* result from a given set of circumstances." *Cont'l Can*, 948 F.2d at 1269 (emphasis in original). As discussed above, the Oracle 2.0 References only disclose that the WRB Dispatcher narrows the universe of WRBXs capable of processing a request based upon: (1) the request; (2) a configuration file; and (3) whether a WRBX is already processing another request. The Oracle 2.0 References, however, fail to disclose how the WRB Dispatcher chooses between the WRBXs capable of processing the request and that are not already processing another request. Thus, the Oracle 2.0 References are completely consistent with the WRB Dispatcher making an uninformed selection of a WRBX ("page server"), *e.g.*, first available, round-robin, or random. Thus, Oracle 2.0 References do not disclose the "dispatching" limitation under epicRealm's construction because they fail to disclose making an "informed selection" of a "page server."

ii. The Oracle 2.0 References Do Not Disclose "Dispatching" Under Oracle's Construction

Oracle's construction for "dispatching" is "analyzing a request to make an informed selection of which page server should process the request, and sending the request to that page server." As discussed above, the Oracle 2.0 References do not disclose making an "informed

selection" as required by Oracle's construction because there is no disclosure of how the WRB Dispatcher chooses between the idle WRBXs capable of processing a request and Oracle concedes that, for example, a random selection would not be an "informed selection." (Exh. 27, Shamos Dep. 171:19-172:17.) Further, the WRB Dispatcher does not make an "informed selection" because it does not distinguish between the relative efficiencies of the WRBXs (page servers) that can process a request as Oracle says is required under its construction. (Exh. 23, Clark Dep. 101:25-104:6.)

c. The Oracle 2.0 References Do Not Disclose "Intercepting" Under Either Party's Construction.

The Oracle 2.0 References do not disclose the "intercepting" limitation of the asserted claims under either party's construction because the "Web server" of the Oracle 2.0 References merely routes a request to a "page server" without examining it and without making a determination of whether the "Web server" or a "page server" should process the request. In fact, as shown below, the Oracle 2.0 References actually disclaim "intercepting" a request.

REDACTED

Indeed, the file history for the '335 Patent shows that "merely routing a request from a Web server to the [page server] does not involve interception." (Exh. 22 at EPIC000498.) In the Oracle 2.0 References, "[t]he HTTP engine does not make any attempt at interpreting the [request], instead it is immediately handed off to the Web Request Broker [Dr. Shamos's 'dispatcher'] for further processing." (Exh. 8 at ORCL000766.) Thus, the Oracle 2.0 References do not disclose the "intercepting" limitation of the asserted claims.

Furthermore, the Oracle 2.0 References actually teach away from "intercepting." Specifically, Reference 604 states that: "[t]he typical HTTP API enables a custom extension to be linked into the HTTP server process and allows it to ***intercept*** and fulfill a client request." (Exh. 8 at ORCL000764 (emphasis added).) Oracle 2.0, however, does not implement the "typical HTTP API" which intercepts a request because:

> the problem with the HTTP API approach is that all extensions are linked into the same executable and thus share the same memory, process, and signal handling space. ... Oracle's answer to these issues is a new and improved architecture for the server, which effectively layers Oracle Web Server 2.0 into two fundamental components: a HTTP engine (the Oracle Web Listener) and a high-speed dispatch mechanism (the Web Request Broker) which routes requests to server extensions running in separate processes.

(Exh. 8 at ORCL000764-65.) Thus, instead of implementing the "typical API," in the Oracle 2.0 References, the "Web server" immediately forwarded a request to the "dispatcher" without determining if a "page server" or the "Web server" should process the request. Hence, the Oracle 2.0 References teach away from "intercepting." Therefore, the Oracle 2.0 References do not anticipate any of the asserted claims for the additional reason that they fail to disclose the "intercepting" limitation.

In sum, the Oracle 2.0 References do not anticipate any of the asserted claims because they do not disclose the "dispatching" or "intercepting" limitations under either party's constructions. In addition, the Oracle 2.0 product cannot anticipate the asserted claims because Oracle merely relies on the other Oracle 2.0 References as evidence of the operation of the Oracle 2.0 product.

V. CONCLUSION

For the foregoing reasons, the Court should grant epicRealm's motion for partial summary judgment that the Oracle References do not anticipate any of the asserted claims.

POTTER ANDERSON & CORROON LLP

OF COUNSEL:

Harry J. Roper
George S. Bosy
Aaron A. Barlow
Patrick L. Patras
David R. Bennett
Paul D. Margolis
Benjamin J. Bradford
Emily C. Johnson
JENNER & BLOCK
330 N. Wabash Avenue
Chicago, IL 60611-7603
Tel: (312) 923-8305

By: */s/ David E. Moore*

Richard L. Horwitz (#2246)
David E. Moore (#3983)
Hercules Plaza, 6th Floor
1313 N. Market Street
Wilmington, DE 19899
Tel: (302) 984-6000
rhorwitz@potteranderson.com
dmoore@potteranderson.com

Attorneys for Defendant/Counterclaim Plaintiff epicRealm Licensing, LP

Dated: July 31, 2008
876828 / 31393 / Oracle

Public Version Dated: August 7, 2008

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

CERTIFICATE OF SERVICE

I, David E. Moore, hereby certify that on August 7, 2008, the attached document was electronically filed with the Clerk of the Court using CM/ECF which will send notification to the registered attorney(s) of record that the document has been filed and is available for viewing and downloading.

I further certify that on August 7, 2008, the attached document was Electronically Mailed to the following person(s):

Mary B. Graham
Morris, Nichols, Arsht & Tunnell LLP
1201 North Market Street
P.O. Box 1347
Wilmington, DE 19899
MGraham@MNAT.com

James G. Gilliland
Igor Shoiket
Townsend and Townsend and Crew LLP
Two Embarcadero Center
Eighth Floor
San Francisco, CA 94111-3834
OracleEpicrealm@townsend.com

Theodore T. Herhold
Robert J. Artuz
Eric M. Hutchins
Eric A. Mercer
Joseph A. Greco
Nitin Gupta
Townsend and Townsend and Crew LLP
379 Lytton Avenue
Palo Alto, CA 94301
OracleEpicrealm@townsend.com

Chad E. King
Townsend and Townsend and Crew LLP
1200 Seventeenth Street
Suite 2700
Denver, CO 80202
OracleEpicrealm@townsend.com

/s/ David E. Moore
Richard L. Horwitz
David E. Moore
Potter Anderson & Corroon LLP
Hercules Plaza – Sixth Floor
1313 North Market Street
P.O. Box 951
Wilmington, DE 19899-0951
(302) 984-6000
rhorwitz@potteranderson.com
dmoore@potteranderson.com