IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

ORACLE CORPORATION and ORACLE U.S.A., INC.,	)	
Plaintiffs-Counterdefendants,	)	
v.	)	C.A. No. 06-414 (SLR)
EPICREALM LICENSING, LP,	)	**REDACTED PUBLIC VERSION**
Defendant-Counterclaimant.	)	

ORACLE'S OPPOSITION TO EPICREALM'S MOTION FOR PARTIAL SUMMARY JUDGMENT THAT THE ORACLE REFERENCES DO NOT ANTICIPATE

MORRIS, NICHOLS, ARSHT & TUNNELL LLP
Mary B. Graham (#2256)
James W. Parrett, Jr. (#4292)
1201 N. Market Street
P.O. Box 1347
Wilmington, DE 19899-1347
(302) 658-9200
mgraham@mnat.com
jparrett@mnat.com
Attorneys for Oracle Corporation and Oracle U.S.A., Inc.

OF COUNSEL:

James G. Gilliland
Theodore T. Herhold
Joseph A. Greco
Robert J. Artuz
Eric M. Hutchins
TOWNSEND AND TOWNSEND AND CREW LLP
379 Lytton Avenue
Palo Alto, CA 94301
(650) 326-2400

Dorian Daley
Peggy E. Bruggman
Matthew M. Sarboraria
ORACLE CORPORATION
ORACLE U.S.A., INC.
500 Oracle Parkway
Redwood Shores, CA 94065

DATED: August 21, 2008
Redacted Filing Date: August 28, 2008

TABLE OF CONTENTS

Page

I. NATURE AND STAGE OF THE PROCEEDINGS 1

II. SUMMARY OF ARGUMENT 1

III. STATEMENT OF FACTS 2

A. EpicRealm Mischaracterizes the Prior Art and the Problems that the Claimed Inventions Purported to Solve 2

1. The Patents-in-Suit State that the Main Problem with Prior Art Systems Was Not Specifically CGI, but Increasing and Excessive Processing Load on Web Servers 2

2. Contrary to What the Patents-in-Suit Represent, Prior Art Systems Used CGI to Dispatch Requests to Third-Tier Page Servers 5

3. Even EpicRealm's Paid Expert Consultant Believed that Prior Art Systems that Used CGI Rendered the Patents-in-Suit Unenforcable 6

B. The Named Inventors' Purported Solution to the Prior Art's Two-Tier Architecture 7

1. Intercepting the Dynamic Web Page Request at the Web Server 8

2. Dispatching the Dynamic Web Page Request to a Page Server 10

C. Beginning in 1994, Oracle Developed Its Own Prior Art Web Server Products Including Oracle WebServer 1.0 and 2.0 11

1. Oracle's Development of Oracle WebServer 1.0 11

2. Oracle's Development of Oracle WebServer 2.0 13

D. EpicRealm's Accusations Regarding Oracle's Accused Products are Erroneous 15

IV. LEGAL STANDARDS 15

A. Summary Judgment 15

B. Invalidity Under 35 U.S.C. § 102 – Anticipation 16

V. ARGUMENT 16

A. The Oracle References Are All Prior Art Under 35 U.S.C. 102(a) 16

1. EpicRealm Cannot Prove an Invention Date Prior to the Filing Date of the EpicRealm Patents-in-Suit....16

2. The Oracle References Predate EpicRealm's April 23, 1996 Invention Date....19

3. OWS 1.0 Is Not Cumulative to any Prior Art that Was Before the Patent Office During its Initial Examination of the Asserted Claims....21

B. Oracle WebServer 1.0, as Disclosed in the Oracle References, Anticipates the Asserted Claims Under 35 U.S.C. §§ 102(a) and (g)....22

1. The Oracle WebServer User's Guide (Reference 153) Discloses "Dispatching"....24

a) The Oracle WebServer User's Guide Discloses "Dispatching" Under EpicRealm's Proposed Construction....24

b) The Oracle WebServer User's Guide Discloses "Dispatching" Under Oracle's Proposed Construction....26

2. The "Oracle WebServer: A Technical Discussion" Document (Reference 503) Discloses Dispatching Under Both Parties' Proposed Constructions....27

3. The SQL*Net References Discloses Dispatching Under Both Parties' Proposed Constructions....28

a) The SQL*Net 2.2 Reference Discloses "Dispatching"....28

b) The SQL*Net 2.3 Reference Discloses "dispatching"....32

4. The OWS 1.0 Product Discloses "Dispatching" Under Both Parties' Proposed Constructions....32

5. Oracle Can Combine Reference 153 with the SQL*Net References for Purposes of Proving Anticipation of the Asserted Claims....33

C. Oracle WebServer 2.0, as Disclosed in the Oracle References, Anticipates the Asserted Claims Under 35 U.S.C. §§ 102(a) and (g)....34

1. The 605 Reference and the OWS 2.0 User's Guide Disclose Both "Intercepting" and "Releasing"....36

2. Oracle WebServer 2.0 Discloses Dispatching Under Both Parties' Proposed Constructions....36

3. Oracle WebServer 2.0, as Disclosed in the Oracle References, Discloses Intercepting Under Both Parties' Proposed Constructions 39

VI. CONCLUSION 40

TABLE OF AUTHORITIES

Page

Cases

Advanced Display Sys. Inc. v. Kent State Univ.
212 F.3d 1272 (Fed. Cir. 2000) 33

Anderson v. Liberty Lobby, Inc.
477 U.S. 242 (1986) 16

Armco, Inc. v. Cyclops Corp.
791 F.2d 147 (Fed. Cir. 1986) 15

Brown v. 3M
265 F.3d 1349 (Fed. Cir. 2001) 16

Chen v. Bouchard
347 F.3d 1299 (Fed. Cir. 2003) 17

Cooper v. Ford Motor Co.
748 F.2d 677 (Fed. Cir. 1984) 16, 32

Mahurkar v. C.R. Bard, Inc.
79 F.3d 1572 (Fed. Cir. 1996) 17

Maldonado v. Ramirez
757 F.2d 48 (3d Cir. 1985) 2

N.L.R.B. v. FES
301 F.3d 83 (3d Cir. 2002) 2, 17, 31

Novartis Corp. v. Ben Venue Labs., Inc.
271 F.3d 1043 (Fed. Cir. 2001) 15

P&G v. Teva Pharms. USA, Inc.
536 F. Supp. 2d 476 (D. Del. 2008) 16

Pa. Coal Ass'n v. Babbitt
63 F.3d 231 (3d Cir. 1995) 15

Phillips v. AWH Corp.
415 F.3d 1303 (Fed. Cir. 2005) 23

Scripps Clinic & Research Found. v. Genentech, Inc.
927 F.2d 1565 (Fed. Cir. 1991) 16

Telemac Cellular Corp. v. Topp Telecom, Inc.
247 F.3d 1316 (Fed. Cir. 2001) 16

Statutes

35 U.S.C. §§ 102(a) and (g) ..... passim

I. NATURE AND STAGE OF THE PROCEEDINGS

Plaintiffs and Counterdefendants Oracle Corporation and Oracle U.S.A. Inc. (jointly "Oracle") submit the following opposition to Defendant and Counterclaimant epicRealm Licensing LP's ("epicRealm") Motion for Partial Summary Judgment that the Oracle References Do Not Anticipate.

Oracle incorporates by reference its summary of the Nature and Stage of Proceedings set forth in its Opening Claim Construction Brief (D.I. 203), filed on July 31, 2008. EpicRealm asserts in this case that Oracle infringes claims 1-5 and 7-11 of U.S. Patent No. 5,894,554 and claims 2 and 16 of U.S. Patent No. 6,415,335 (the "asserted claims" of the "patents-in-suit"). Oracle asserts that the asserted claims are invalid as anticipated or obvious in view of many prior art references including the Oracle prior art references that are the subject of epicRealm's motion.

II. SUMMARY OF ARGUMENT

1. Although it claims an invention date of November 1995, epicRealm has not met its burden of proving an invention date prior to the April 23, 1996 filing date of its patents.

2. The Oracle References, including the Oracle WebServer 1.0 and 2.0 products and the documentation that describes them, are prior art to the patents-in-suit under 35 U.S.C. §§ 102(a) and (g) because they were conceived and diligently reduced to practice, publicly available, on sale, or in public use prior to epicRealm's actual invention date of April 23, 1996 and, in many cases, prior to epicRealm's asserted invention date of November 1995.

3. The asserted claims of the patents-in-suit are invalid under 35 U.S.C. §§ 102(a) and (g) because the Oracle References disclose each limitation of every asserted claim. EpicRealm's summary judgment motion focuses only on the allegation that the Oracle References fail to disclose the "dispatching" and "intercepting" claim limitations for particular references. EpicRealm has failed to meet its burden of proof with respect to other claim

limitations. EpicRealm's arguments fail because epicRealm misstates the facts with respect to the prior art and improperly imports limitations into the parties' proposed construction for "dispatching" in an effort to evade invalidating prior art. Additionally, many of epicRealm's arguments simply amount to unsupported, conclusory assertions, which are inadequate to satisfy epicRealm's burden of proof on summary judgment. *See, e.g.*, *N.L.R.B. v. FES*, 301 F.3d 83, 95 (3d Cir. 2002); *Maldonado v. Ramirez*, 757 F.2d 48, 51 (3d Cir. 1985).

4. EpicRealm has not met its initial burden on summary judgment of showing an absence of a genuine issue of material fact. Even if it were found that epicRealm has met its initial burden, Oracle has presented substantial evidence in its Motion for Summary Judgment of Invalidity and in this Opposition showing that summary judgment of invalidity should be granted in Oracle's favor or, at a very minimum, that there exists genuine issues of material fact that preclude summary judgment in epicRealm's favor.

III. STATEMENT OF FACTS

The Statement of Facts in Oracle's opening brief on Claim Construction filed on July 31, 2008 (D.I. 203) contains background on the parties and general facts of the case.

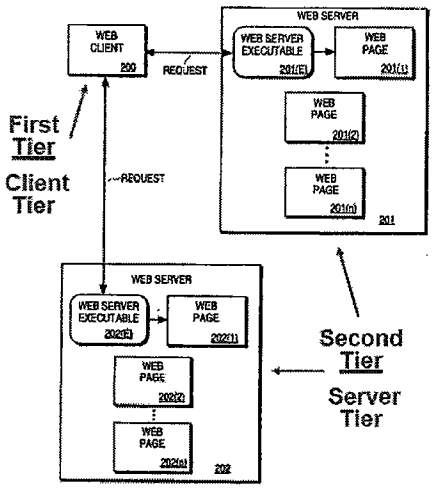
A. EpicRealm Mischaracterizes the Prior Art and the Problems that the Claimed Inventions Purported to Solve

1. The Patents-in-Suit State that the Main Problem with Prior Art Systems Was Not Specifically CGI, but Increasing and Excessive Processing Load on Web Servers

EpicRealm spends significant time trying to distinguish its claimed inventions from early prior art systems that used the Common Gateway Interface ("CGI") to generate dynamic Web pages. This is because epicRealm is understandably concerned that many early prior art systems that utilized CGI to manage dynamic Web page requests anticipate the asserted claims of the patents-in-suit. EpicRealm cannot distinguish its claimed inventions from prior art systems that

utilized CGI because ***not a single limitation of the asserted claims would preclude the use of CGI to manage dynamic Web page requests.***

In an effort to distinguish the claimed inventions from CGI, epicRealm mischaracterizes the primary problem in the prior art that the claimed inventions purported to solve. This problem was not specifically CGI; rather, it was the increasing load on Websites as a result of the 1995-96 Internet boom. A1[1] at col. 2:1-12 and 4:33-53. The patents-in-suit state that the major problem facing Websites at the time was that Web servers were getting hit with numerous Web page requests, and the server machines had to handle the heavy processing of all the requests because they had no way to hand off some of the load to other servers. *Id.* at col. 4:11-53. In other words, the patents-in-suit represented that the prior art was limited to a two-tier client-server architecture, where a client located in a first tier would send requests to one or more servers located in a second tier which handled all the processing for the client. The patents-in-suit illustrate this alleged problem in Fig. 2 of the specification:



[1] Appendix Exhibits 1-110 are contained in Oracle's Appendix of Exhibits in Support of Oracle's Opening Claim Construction Brief and Motions Filed on July 31, 2008 and in Oracle's Supplemental Appendix of Oracle Exhibits Filed on August 21, 2008.

Id. at Fig. 2 ("First Tier" and "Second Tier" labels added).

Fig. 3 of the patents-in-suit further illustrates the prior art Web server environment of Fig. 2 in the form of a flow diagram. *Id.* at col. 4:11-12 and Fig. 3. Using the flow diagram of Fig. 3, the patents-in-suit state that the Web servers located in the second tier of the prior art systems used CGI applications to process all dynamic Web page requests on the same Web server:

> In processing block 304 the request is then transmitted from the Web browser to the appropriate Web server, and in processing block 306 the Web server executable examines the URL to determine whether it is a HTML document or a CGI application.
>
> * * *
>
> If the URL request is for a CGI application 314 . . . the Web server executable locates the CGI application in processing block 316. The CGI application then executes and outputs HTML output in processing block 318 and finally, the HTML output is transmitted back to requesting Web browser for formatting and display in processing block 320.

Id. at col. 4:16-31. The patents-in-suit further state that the real problem with the prior art was not specifically CGI, but that prior art systems were limited to two-tier architectures where each Web server was responsible for handling all of the processing of Web page requests on a ***single Web server machine*** containing a single processor:

> For example, a large Web site may receive thousands of requests or "hits" in a single day. ***Current Web servers process each of these requests on a single machine, namely the Web server machine.*** Although these machines may be running "multi-threaded" operating systems that allow transactions to be processed by independent "threads," ***all the threads are nevertheless on a single machine, sharing a processor. As such, the Web executable thread may hand off a request to a processing thread, but both threads will still have to be handled by the processor on the Web server machine.*** When numerous requests are being simultaneously processed by multiple threads on a single machine, the Web server can slow down significantly and become highly inefficient. The claimed invention addresses this need by utilizing a partitioned architecture to facilitate the creation and management of custom Web sites and servers.

Id. at col. 4:38-53 (emphasis added). Accordingly, the main problem that the claimed inventions attempted to solve was the limitations of the two-tier architecture, namely, the disadvantages of

having the server tier do all the work of managing the incoming requests. The patents-in-suit further state that the prior art needed a new method and apparatus for managing dynamic web page requests which reduced the processing load on the Web servers so that the Web servers could more efficiently manage dynamic Web page requests. *Id.* The claimed inventions purported to solve this problem by utilizing a "partitioned architecture," *i.e.*, a multi-tier architecture in which a third tier comprised of "page servers" was added to offload dynamic Web page requests from the Web server. *Id.* at col. 4:51-53.

2. Contrary to What the Patents-in-Suit Represent, Prior Art Systems Used CGI to Dispatch Requests to Third-Tier Page Servers

EpicRealm tries to blame the Web server load problem on CGI, but the prior art's use of CGI was not the main cause of this problem. The patents-in-suit even admit that the real problem with the prior art was the two-tier architecture, in which the Web servers were tasked with handling all of the dynamic Web page requests. A1 at col. 4:38-53. CGI did not require a Web server to process all requests on the same machine. CGI ("common gateway interface"), as its full name suggests, is simply a standard protocol for interfacing a Web server with special application software. A83 [Shamos Dep.] at pp. 264:19-266:11; A19 [Chen Depo.] at pp. 69:14-74:14 and Exh. 5 at p. 3. A Web server could, as the patents-in-suit note, use CGI to implement numerous CGI applications on the same Web server machine. A83 at pp. 267:2-23; A19, Exh. 5 at p. 3. Such an unsophisticated use of CGI could, of course, severely bog down a Web server tasked with handling numerous Web page requests. *See* A83 at pp. 267:2-23. Prior art Web servers, however, were not restricted to using CGI to implement applications on the same Web server machine. *See* A83 at pp. 266:14-267:1; A19, Exh. 5 at pp. 4-6.

The patents-in-suit fail to recognize that the industry had already figured out how to use CGI and similar techniques to do exactly what the claimed inventions describe. Particularly, the

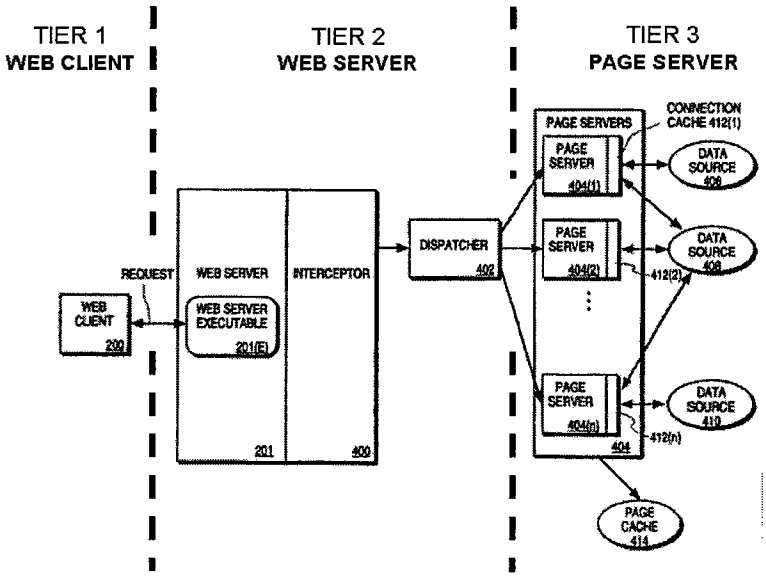
prior art included many systems that were using CGI to dispatch dynamic Web page requests from a Web server machine to a third-tier page server machine for processing. *See* A83 at pp. 263:12-21 and 266:14-267:1; A19, Exh. 5 at pp. 4-6; *see also* Declaration of Michael Shamos in Support of Oracle's Opposition to EpicRealm's Motions ("Shamos Opp. Decl.") at ¶¶29-37 and 51-52. Some of those prior art systems are discussed in Oracle's Motion for Summary Judgment of Invalidity of the Asserted Claims including Oracle's WebServer 1.0, NeXT Software Inc.'s Web Objects, and Cornell University's Dienst Server. D.I. 207. All of these systems could use CGI to practice the claimed "partitioned architecture" of the patents-in-suit, and all such systems anticipate the asserted claims. *Id.*



[redacted]

B. The Named Inventors' Purported Solution to the Prior Art's Two-Tier Architecture

The alleged "invention" that the named inventors came up with to solve the efficiency problems associated with the two-tier client-server architecture was the addition of a third tier (a "page server" tier) to which the Web server could offload requests for processing. This three-tier architecture of the patents-in-suit is illustrated in Fig. 4 of the patents-in-suit:



A1 at Fig. 4 (dotted vertical lines and "Tier" labels added).

The independent claims of the patents-in-suit are generally directed to a method (or an apparatus for practicing a method) for handling dynamic Web page requests including (1) intercepting a request at a Web server, (2) dispatching that request to a particular page server, (3) releasing the Web server to process other requests, and (4) generating the requested dynamic Web page. *See, e.g., id.* at claim 1. As discussed in the patents' specification, the goal of the claimed inventions is to offload the processing of dynamic Web page requests from the Web server to software running on a separate computer (the page server), thus reducing the processing load on the Web server. *Id.* at col. 4:33-54. This concept, however, of spreading out server processing across multiple computers situated across separate tiers was well known in the art even before the Internet boom of 1995. Shamos Opp. Decl. at ¶¶29-37. Additionally, the claimed step of "dispatching" a request to a particular server (*e.g.*, a page server), selected from among several servers, was also well known. *Id.* at ¶¶35-42.

At the time of the asserted claims' alleged invention, the Internet community, including Oracle, had already devised identical solutions to the problems purported to be solved by the patents' applicants. *Id.* at ¶¶40-45. Under epicRealm's application of the parties' claim constructions, these early solutions anticipated each and every asserted claim.

1. Intercepting the Dynamic Web Page Request at the Web Server

In the claimed inventions, requests for dynamic content are identified and "intercepted at the Web server." The Web server software (Web server executable) does not process these dynamic Web page requests. With reference to Figure 4, the patent states:

> Web client 200 issues a URL request that is processed to determined proper routing. In this embodiment, the request is routed to Web server 201. Instead of Web server executable 201(E) processing the URL request, however, Interceptor 400 intercepts the request and routes it to Dispatcher 402.

Col. 4:55-60. There is no disclosure or indication in the patent or the claims that the Web server will process any request for dynamic content; these requests are "intercepted" so that the processing is done elsewhere. The only Web page requests that the Web server may process, therefore, are Web page requests for static content.

EpicRealm's description of the intercepting requirement is based on an "example" of the Web server processing static requests and not processing dynamic requests:

> 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 the Dispatcher 402." 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" . . . .

ER Br. at 5 (citations omitted) (emphasis added). EpicRealm's use of the term "may" attempts to leave open the possibility that (1) static Web page requests could be processed by some part of the system other than the Web server, and (2) dynamic Web page requests could be processed by the Web server.

There is no indication in the patent or the claims, however, that "static Web page requests" will be sent to a page server for processing. To the contrary, the claimed "interception" of requests for dynamic content at the Web server must include identifying dynamic requests, *as opposed to static requests*, and diverting them from the Web server. Similarly, there is no indication in the patent or the claims that "dynamic Web page requests" will be processed by the Web server. The only disclosure in the patent and the claims is that requests for dynamic content will be "intercepted at the Web server" and processed by a "page server."

EpicRealm argues that "[w]hen the request is received at the Web server, the Web server

initially has the duty to process the request." ER Br. at 6. Nevertheless, both the specification and the claims require that the Web server ***not*** have the "duty" to complete processing requests for dynamic content, but that these requests be "intercepted" and processed by a "page server." As discussed below, both Oracle's WebServer 1.0 and 2.0 products distinguish between requests for static Web pages and requests for dynamic Web pages and perform intercepting according to the claimed inventions. *See* Section V.C., *infra*.

2. Dispatching the Dynamic Web Page Request to a Page Server

After a dynamic Web page request is intercepted at the Web server, it is routed to a dispatcher. *See* Dispatcher 402 of Fig. 4 of the '554 patent [A1]. The dispatcher examines (or analyzes) the dynamic Web page request to make an informed selection of which page server should process the request. *Id.* at col. 5:38-53. As stated in the '554 patent: "Dispatcher 402 maintains a variety of information regarding each Page server on the network, and dispatches requests based on this information." *Id.* at col. 5:51-53. Accordingly, the dispatcher relies on information in the "URL request" (*e.g.*, directory information associated with a particular database) and information about the page servers (*e.g.*, which page servers have access to the particular database) to select a page server than can process the request.

While the patents-in-suit offer examples of how the dispatcher may examine a request to make an informed section of which page server should process the request, epicRealm seeks in its claim construction to import limitations described in those examples into the meaning of "dispatching." *See, e.g.,* ER Br. at p. 6. The several dispatching examples discussed in the '554 patent, however, are merely alternate embodiments of the claimed invention and should not serve to limit the broad meaning of "dispatching." A1 at col. 8:58-61.

EpicRealm, for example, states that the dispatcher will always route to a pager server that can more efficiently process the request based on dynamic information maintained about the

pager servers. ER Br. at p. 6. Again, these limitations are described as exemplary embodiments of the claimed invention and not as mandatory limitations for every claimed invention. A1 at col. 5:49-6:65. For example, there is no requirement in the specification that the dispatcher *always* rely on *dynamic* information about page servers to make an informed selection. There is nothing in the specification that precludes the dispatcher from relying on static information (*e.g.*, information that does not change during the software's execution) to make an informed selection of which page server should process the request. *See* Oracle's Opening Claim Construction Brief (D.I. 203) at pp. 27-32.

EpicRealm further states that "dynamic information": (1) "is information that can only be established during the execution of a program," (2) "is not information that can be established prior to the execution of the program," and (3) "must indicate which page server can more efficiently process the request." ER Br. at pp. 6-7. There is no support in the specification for all these limitations. One of ordinary skill in the art would not understand "dynamic information" to include all these extraneous limitations. A83 at pp. 129:24-131:9.

C. Beginning in 1994, Oracle Developed Its Own Prior Art Web Server Products Including Oracle WebServer 1.0 and 2.0

Founded in 1977 by Silicon Valley software engineers Larry Ellison, Bob Miner, and Ed Oates, Oracle specializes in developing and marketing enterprise software products, particularly database management systems, middleware including internet application servers, and business applications. In 1995, Oracle was no stranger to the Internet. Oracle was, to the contrary, a pioneer in integrating World Wide Web innovation with database technology to make information management more powerful than ever.

1. Oracle's Development of Oracle WebServer 1.0

In early 1995, Oracle engineers conceived of a sophisticated web-based product called

Oracle WebServer 1.0 ("OWS 1.0"). Declaration of Katrina Montinola in Support of Oracle's Opposition to EpicRealm's Motion for Summary Judgment ("Montinola Opp. Decl.") at ¶4. OWS 1.0 together with the then well-known Oracle7 Server (a database server) comprised a system for providing access to database information over the Internet. *Id.* As shown in Fig. A below, OWS 1.0 included a Web Listener for interfacing with a client (*e.g.*, a web browser) and a Web Agent for dispatching dynamic Web page requests to one or more Oracle7 Servers. *Id.*

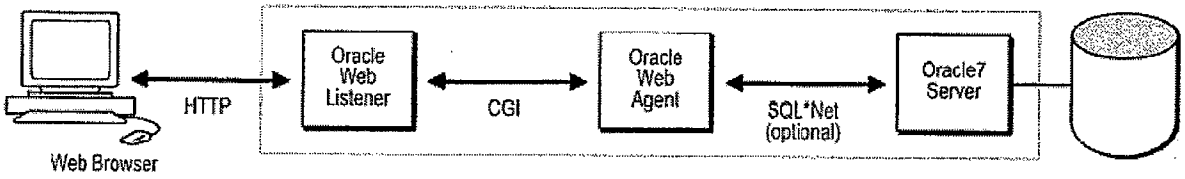


Figure A - from the OWS 1.0 User's Guide - A40 at ORCL000022

Client requests are received at the Web Listener, which is a web server capable of handling requests for both static and dynamic web pages. *Id.* Requests for dynamic web pages are intercepted at the Web Listener and are routed to the Web Agent. *Id.* The Web Agent examines the request to make an informed selection of which database server (or which application on the database server) should process the request.[2] *Id.* The Web Agent then logs into the appropriate Oracle7 Server and dispatches the request to the appropriate application for the generation of the requested dynamic Web page. *Id.*

Oracle worked diligently to reduce OWS 1.0 to practice by July 1995, and the beta version of the product was made accessible to the public in September 1995, well before the

---

[2] EpicRealm tries to distinguish OWS 1.0 from the claimed inventions by arguing that the Web Agent does not have a "choice" as to which page server it must send the request. EpicRealm's argument, however, is nonsensical because it attempts to assign human qualities to a software program. The dispatcher of the claimed inventions does not have a choice either. It is forced to select a particular page server based on the information it relies on and how it is programmed to make its selection based on that information. It has no "choice" in the matter.

earliest invention date promulgated by epicRealm. *Id.* at ¶¶5-7. The beta release represented a finished product, and its core architecture and operation were identical to that of the first commercial release of OWS 1.0 which occurred in October 1995. *Id.* at ¶¶6-8.

OWS 1.0 was designed to be used with the Oracle7 Server, which was Oracle's database product. *Id.* at ¶4. OWS 1.0 used the Oracle7 Server to generate dynamic Web pages. *Id.* OWS 1.0 could connect to the Oracle7 Server directly or using a feature called SQL*Net. SQL*Net allowed the WebServer to connect to any of a variety of Oracle7 Servers over a network. Shamos Opp. Decl. at ¶¶53-54. Contrary to what epicRealm argues in its brief, the Oracle Web Agent together with SQL*Net could implement dynamic load balancing to select a particular page server to process a dynamic Web page request.[3] *Id.* at ¶¶55-56.

2. Oracle's Development of Oracle WebServer 2.0

While Oracle was developing OWS 1.0 it was concurrently developing its next generation product, OWS 2.0. Montinola Opp. Decl. at ¶10. OWS 2.0 operated similarly to OWS 1.0 but also included a few enhancements.[4] For example, like OWS 1.0, the OWS 2.0 Web server "listens for incoming HTTP requests, delivers static files and runs simple CGI

---

[3] EpicRealm argues that in 1995 Oracle was not aware of any Web server that used load balancing. ER Br. at pp. 10-11. But even the Web server of the claimed inventions could not independently perform load balancing. EpicRealm only asserts that the *dispatcher* is capable of performing load balancing. SQL*Net, like the dispatcher of the claimed inventions, is neither a Web server nor a page server, but it could use load balancing to select a page server.

[4] EpicRealm consistently suggests that Oracle "abandoned" its products just because it later released new product versions containing additional features and enhancements. ER Br. at pp. 9-12. This is not accurate. For example, Oracle didn't abandon OWS 1.0 just because it later released OWS 2.0. Oracle continued to distribute OWS 1.0 with its Oracle7 Server even after OWS 2.0 had been released, and OWS 2.0 retained many of OWS 1.0's features including CGI and the PL/SQL Web Agent. A42 at ORCL01816651-654 (describing CGI). Similarly, Oracle didn't abandon OWS 2.0. Again, it just released an improved version of the same product, OWS 3.0, which retained many of OWS 2.0's core features. *See* A44 at ORCL000780.

programs." A44 at ORCL000764. Additionally, as show below in Fig. B, the OWS 2.0 Web server included the new ability to intercept dynamic Web page requests and hand them off to a Web Request Broker for special handling. *Id.*; A42 at ORCL01816652.

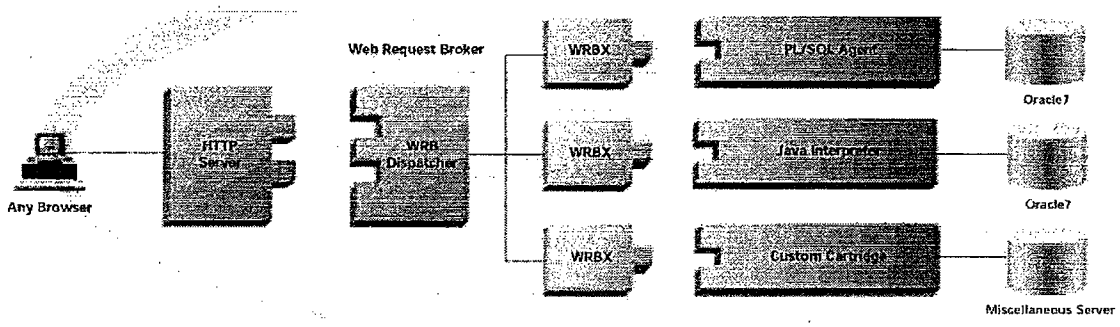


Figure B – A43 at ORCL000747

The Web Request Broker included a Dispatcher that dispatched dynamic Web page generation requests to processes known as WRBXs, which are instances of WRB Services. A42 at ORCL01816652-653. Each WRBX interfaced with one of at least three different WRB Services including a PL/SQL Agent, a Java Interpreter, and a LiveHTML Interpreter. *Id.* The WRB Dispatcher dispatched requests to WRBXs based on dynamic information including (1) what WRB Service was called for by the request; (2) which WRBXs are connected with the requested WRB Service, and (3) load information concerning the various WRBXs. The Dispatcher's ability to use load balancing to select appropriate software to generate the requested Web page is described in an early version of the OWS 2.0 User's Guide:

> The Dispatcher continually adjusts the load by controlling how many WRBX's are running at a given time, subject to certain parameters. These parameters are set by the WebServer Administrator, who decides the maximum and minimum number of WRBX's running. The Dispatcher creates new WRBX's as needed and connects them to the appropriate WRB Services. The Dispatcher keeps track of which WRBX's are executing requests and which are free, and periodically checks the WRBX's to see how long they have been idle. If a WRBX's idle time is beyond the maximum set by the WebServer Administrator, that WRBX is killed. The WRBX's assigned to a given WRB Service always have requests assigned in the same order, so that traffic will be concentrated in a small number them, and most of them will not be idle.

A98 at ORCL01607402; *see also* A42 at ORCL01816653.

The core components of OWS 2.0 -- including the Web Listener (the Web server), the Web Request Broker (the dispatcher), and the WRB Services (the page servers) -- were conceived by Oracle no later than July 1995, and Oracle worked diligently to reduce OWS 2.0 to practice by October 1995, again well before epicRealm's alleged invention date. Montinola Opp. Decl. at ¶¶11-12. Later, in February 1996, Oracle disclosed OWS 2.0 to the public at an E-mail World & Internet Expo in San Jose, California. *Id.* at ¶¶13-14.

D. EpicRealm's Accusations Regarding Oracle's Accused Products are Erroneous

EpicRealm makes a number of inaccurate accusations about Oracle and its accused products. ER Br. at pp. 14-15. Because these accusations are not relevant to whether Oracle's prior art products anticipate the asserted claims, Oracle will not address them here. EpicRealm's infringement accusations are instead addressed in Oracle's Motions for Summary Judgment of Noninfringement and No Willful Infringement (D.I. 205 and 209) and in Oracle's Opposition to EpicRealm's Motion for Partial Summary Judgment of Literal Infringement, filed herewith.

IV. LEGAL STANDARDS

A. Summary Judgment

Summary judgment may only be granted if, after viewing the alleged facts in the light most favorable to Oracle and drawing all justifiable inferences in Oracle's favor, there is no genuine issue whether the Oracle references anticipate the asserted claims. *See Novartis Corp. v. Ben Venue Labs., Inc.*, 271 F.3d 1043, 1046 (Fed. Cir. 2001); *Pa. Coal Ass'n v. Babbitt*, 63 F.3d 231, 236 (3d Cir. 1995). Oracle, as the nonmovant, "is required merely to point to an evidentiary conflict created on the record." *Armco, Inc. v. Cyclops Corp.*, 791 F.2d 147, 149 (Fed. Cir. 1986). Where there is doubt as to the existence of a genuine issue of material fact, that doubt

must be resolved in Oracle's favor. *Cooper v. Ford Motor Co.*, 748 F.2d 677, 679 (Fed. Cir. 1984). In other words, where "there are genuinely disputed issues of material fact, summary judgment cannot be utilized as the tool for deciding those issues." *Armco, Inc.*, 791 F.2d at 151.

Here, there is much more than a "scintilla of evidence" to support invalidity of epicRealm's asserted claims; there is clear and convincing evidence that each of the asserted claims is anticipated by the Oracle prior art (among other references). *See Anderson v. Liberty Lobby, Inc.*, 477 U.S. 242, 252 (1986).

B. Invalidity Under 35 U.S.C. § 102 – Anticipation

In order to anticipate, a single prior art reference must disclose each and every limitation of the claimed invention, either expressly or inherently. *Telemac Cellular Corp. v. Topp Telecom, Inc.*, 247 F.3d 1316, 1327 (Fed. Cir. 2001). There must be no difference between the claimed invention and the reference disclosure as viewed by a person of ordinary skill in the field of the invention. *See Scripps Clinic & Research Found. v. Genentech, Inc.*, 927 F.2d 1565, 1576 (Fed. Cir. 1991). Anticipation is a question of fact for the jury to decide. *Brown v. 3M*, 265 F.3d 1349, 1351 (Fed. Cir. 2001).

V. ARGUMENT

A. The Oracle References Are All Prior Art Under 35 U.S.C. 102(a)

1. EpicRealm Cannot Prove an Invention Date Prior to the Filing Date of the EpicRealm Patents-in-Suit

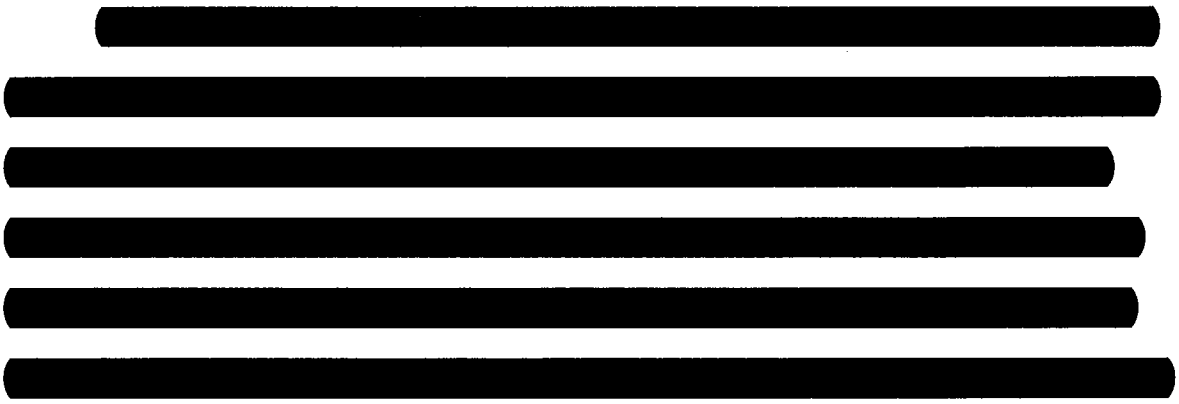
Because the April 23, 1996 filing date of the patents-in-suit is the presumed invention date, epicRealm, "as the party attempting to establish an earlier invention date, bears the burden of rebutting this presumption, by a preponderance of the evidence." *P&G v. Teva Pharms. USA, Inc.*, 536 F. Supp. 2d 476, 490 (D. Del. 2008). Specifically, epicRealm must show that the claimed inventions were conceived and diligently reduced to practice before the publication or

invention date of the prior art. *See Mahurkar v. C.R. Bard, Inc.*, 79 F.3d 1572, 1576-78 (Fed. Cir. 1996); *see also* 35 U.S.C. § 102(g)(2). "When a party seeks to prove conception through oral testimony of a putative inventor, it must offer corroborating evidence." *P&G*, 536 F. Supp. 2d at 491 (citing *Chen v. Bouchard*, 347 F.3d 1299, 1309 (Fed. Cir. 2003)). In assessing corroborating evidence, courts apply a "rule of reason" analysis, in which "an evaluation of all pertinent evidence must be made so that a sound determination of the credibility of the inventor's story may be reached." *Id.* (citation omitted).

EpicRealm contends it is entitled to an invention date of November 1995. ER Br. at p. 21. EpicRealm, however, cannot meet its burden of proving an invention date any earlier than the April 23, 1996 filing date. In an attempt to meet its burden, epicRealm primarily offers the uncorroborated testimony of one of the patents' named inventors, Keith Lowery. [redacted] A76 [Lowery Dep.] at pp. 12:10-13:22. EpicRealm also cites to a number of exhibits that it contends corroborate a November 1995 invention date. EpicRealm's statements, however, are purely conclusory and it has not shown how any of the documents corroborate conception or diligent reduction to practice of the asserted claims. EpicRealm has not met its burden of proof on summary judgment on this issue. *See N.L.R.B.*, 301 F.3d at 95 (A summary judgment movant's evidence "that is essentially conclusory and lacking in specific facts is inadequate to satisfy the movant's burden.") (citation omitted).

As an example of Mr. Lowery's bias in recalling events and dates in a way that favors him and epicRealm, epicRealm contends that a document dated November 15, 1995

(EPIC000146-148) corroborates conception of the claimed inventions in that month.[5] A77 [Levine Dep.] at Exh. 8. But a different named inventor of the patents-in-suit, Andrew Levine, has squarely contradicted epicRealm's claim. Mr. Levine testified at his deposition that the November 15, 1995 document did not show the inventors had conceived of the claimed inventions by that time because the document failed to disclose a dispatcher and page servers, key limitations of all the asserted claims. *Id.* at pp. 104:24:105:16. Accordingly, this document cannot corroborate an invention date because it does not demonstrate that the named inventors had conceived of all the limitations of the claimed inventions. Mr. Levine further testified that he, not Keith Lowery, was the one who later conceived of the dispatcher and page server functionality of the claimed inventions. *Id.* at pp. 110:10-113:1. Mr. Levine also testified that he was not at all surprised that Keith Lowery would try to take all the credit for conceiving of the claimed inventions: "it would not surprise me that Mr. Lowery would claim anything that would further his own ego." *Id.* at pp. 253:13-254:1; *see also* 235:8-236:17.



[5] EpicRealm also refers to a document dated September 15, 1995 as alleged evidence of conception. ER Br. at 21. But epicRealm does not even attempt to claim an invention date back as far as September 1995. EpicRealm again fails to demonstrate how this document supports a conception date earlier than the filing date. EpicRealm has not offered evidence linking the document's disclosure to each limitation of the asserted claims.

[redacted]

2. The Oracle References Predate EpicRealm's April 23, 1996 Invention Date

Oracle has asserted that certain Oracle references anticipate the asserted claims. These references primarily consist of the Oracle WebServer 1.0 and 2.0 products themselves. As discussed above, OWS 1.0 (including the Oracle7 Server) was conceived and diligently reduced to practice prior to August 1995. Montinola Opp. Decl. at ¶¶4-6. OWS 1.0 was also first released to the public in beta form in September 25, 1995 and was later commercially released in October 1995. *Id.* at ¶¶6-8. These dates are corroborated by the documents cited in the Montinola Decl. including A40, A41 and A52-55. Similarly, documents describing these products are also prior art because they each had publication dates no later than the following:

(1) Oracle WebServer User's Guide, Release 1.0 (Reference 153) – October 30, 1995 (Montinola Opp. Decl. at ¶9; A40);

(2) "The Oracle Web Server: A Technical Discussion" publication (Reference 503) – September 25, 1995 (Montinola Opp. Decl. at ¶7; A41);

(3) "SQL*Net: Understanding SQL*Net," Release 2.2 – August 1995 (Montinola Opp. Decl. at ¶6; A54); and

(4) "SQL*Net: Understanding SQL*Net," Release 2.3 – March 1996 (Montinola Opp. Decl. at ¶16; A105 at ORCL01818198).

These references in addition to the OWS 1.0 product (the "OWS 1.0 References") are therefore all prior art pursuant to 35 U.S.C. § 102(a) because they were known, on sale, used, or described in printed publications in the United States prior to epicRealm's actual invention date of April 23, 1996. *See id.* Oracle also relies on the OWS 1.0 product as prior art under 35

U.S.C. § 102(g)(2) because OWS 1.0 (including the Oracle7 Server and SQL*Net (v. 2.2)) was conceived and diligently reduced practice certainly earlier than epicRealm's asserted invention date of November 1995. Montinola Opp. Decl. at ¶¶4-6. OWS 1.0 was also not abandoned, suppressed or concealed because it was commercialized. *Id.* Similarly, SQL*Net (v. 2.3) was reduced to practice prior to epicRealm's actual invention date of April 23, 1996 because it was being sold to customers prior to that month. *Id.* at ¶16.

OWS 2.0 (including the Oracle7 Server) was also conceived and diligently reduced to practice at least prior to November 1995. *Id.* at ¶¶11-12. OWS 2.0 was also first disclosed to the public in February 1996. *Id.* at ¶13. These dates are corroborated by the documents cited in the Montinola Decl. including A42, A44, A51 and A56-58. Similarly, documents describing these products are also prior art because they each had publication dates no later than the following:

(1) "Oracle WebServer 2.0, Technical Note" publication (Reference 605) – March 1996 (Montinola Opp. Decl. at ¶14; A44);

(2) Oracle WebServer's User's Guide, Release 2.0 Production – April 1996[6] (Montinola Opp. Decl. at ¶15; A97);

These references in addition to the OWS 2.0 product (the "OWS 2.0 References") are all prior art pursuant to 35 U.S.C. § 102(a) because they were known, on sale, used or described in printed publications in the United States prior to epicRealm's actual invention date of April 23, 1996. In addition to these references, the Oracle WebServer's User's Guide (Release 2.0.2) also

[6] Oracle contends that the Oracle WebServer's User's Guide, Release 2.0 Production, was publicly available prior to April 23, 1996 because OWS 2.0 was first disclosed to the public by March 1996 and the product's production release would have occurred shortly thereafter, prior to April 23, 1996. Montinola Decl. at ¶¶13-15. At a minimum, the precise date of the document's publication is a question of fact.

disclosed how OWS 2.0 operated as of the date of its public disclosure in February 1996. Montinola Opp. Decl. at ¶15. Oracle also relies on the OWS 2.0 product as prior art under 35 U.S.C. § 102(g)(2) because OWS 2.0 (including the Oracle7 Server) was conceived and diligently reduced practice by October 30, 1995. Montinola Opp. Decl. at ¶¶11-15. OWS 2.0 was also not abandoned, suppressed or concealed because it was commercialized. *Id.*

Contrary to epicRealm's assertions, Oracle never abandoned its theory that the asserted claims are invalid under 35 U.S.C. § 102(g). Oracle made clear in its Fourth Supplemental Response to EpicRealm's Interrogatory No. 3, dated February 15, 2008, that it was asserting invalidity under Section 102(g) based on the OWS 1.0 and 2.0 prior art references. A93 at pp. 3-4. The fact that Oracle's expert, Dr. Shamos, never specifically mentions Section 102(g) in his report is irrelevant, and epicRealm does not cite any authority to the contrary.

3. OWS 1.0 Is Not Cumulative to any Prior Art that Was Before the Patent Office During its Initial Examination of the Asserted Claims

EpicRealm argues that the Oracle References should be ignored because they are merely cumulative to U.S. Patent No. 5,761,673 (the "Bookman patent"), which the Examiner cited during his initial examination of the '554 patent application. EpicRealm is wrong. Although the Bookman patent covers certain aspects of the OWS 1.0, it does not fully disclose all features of the Oracle References which are relevant to the claimed inventions. Indeed, the Bookman patent is only 9 pages long and the OWS 1.0 User's Guide (Reference 153), for example, is over ***220 pages long***. For example, the Bookman patent does not expressly disclose SQL*Net, TCP releasing according to epicRealm's contention, concurrency processing, multiple data sources, connection caching, page caching, and updating a data source. Shamos Opp. Decl. at ¶¶83-84.

Tellingly, epicRealm fails to mention that the Patent Office in fact agrees with Oracle that OWS 1.0 is not cumulative to the Bookman patent. On May 4, 2007, the Patent Office

granted requests to reexamine the claims of the patents-in-suit in view of two Oracle References, finding that such references "raise[d] a substantial new question of patentability" concerning certain claims of the patents-in-suit. *See, e.g.,* A72 at p. 5. The Patent Office recently issued an office action rejecting *all* claims of the '554 patent as anticipated or obvious in view of multiple prior art references. A10. In that office action, the Examiner even cites to the OWS 1.0 User's Guide. *Id.* at pp. 12-16.

B. Oracle WebServer 1.0, as Disclosed in the Oracle References, Anticipates the Asserted Claims Under 35 U.S.C. §§ 102(a) and (g)

As shown in Fig. C below, OWS 1.0 discloses all the limitations of the independent asserted claims including, for example, a Web client (the web browser) sending a dynamic Web page request to a Web server (the Web Listener). A40 at ORCL000021; Shamos Opp. Decl. at ¶48. OWS 1.0 further discloses routing the request from the Web server to a page server (an Oracle7 Server) via a dispatcher (the Web Agent and SQL*Net). Shamos Opp. Decl. at ¶49.

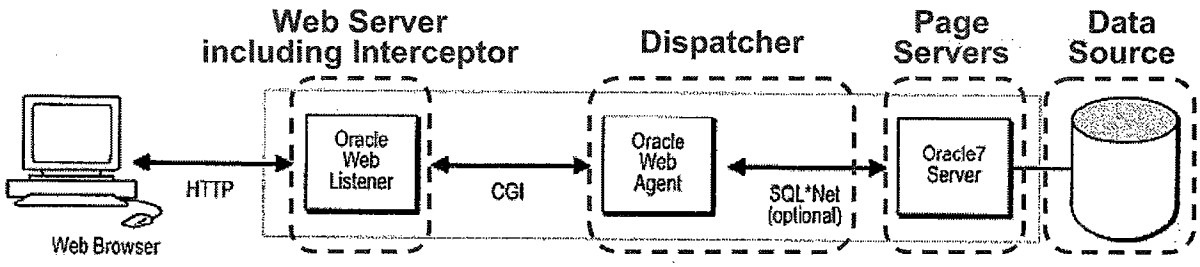


Figure C - from the OWS 1.0 User's Guide - A40 at ORCL000022 (dotted lines and top labels added)

EpicRealm's only anticipation argument with respect to the OWS 1.0 References is that they do not disclose the "dispatching" limitation of the claimed inventions.[7] ER Br. at pp. 22-31.

---

[7] EpicRealm also suggests that Oracle cannot prove that the OWS 1.0 References anticipate the asserted claims based on other reasons. ER Br. at p. 22. EpicRealm, however, fails to raise any other specific argument beside its "lack of dispatching" argument. EpicRealm must be limited to that argument on summary judgment because, as the moving party, it has not met its initial

Continued on the next page

As shown below, epicRealm is wrong because all the Oracle References disclose "dispatching" under both parties' proposed constructions for that term.[8] Particularly, OWS 1.0's Web Agent can dispatch requests, according to the claimed inventions, to a page server by itself or by using a software tool called SQL*Net. Shamos Opp. Decl. at ¶¶52-54.

EpicRealm's lengthy construction for the term "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." As discussed in Oracle's claim construction briefs, epicRealm's construction is improper because it reads in extraneous limitations from the specification. *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1323 (Fed. Cir. 2005) ("[A]lthough the specification often describes very specific embodiments of the invention, we have repeatedly warned against confining the claims to those embodiments."). Alternatively, Oracle's proposed 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." Under either construction, the Oracle References disclose the "dispatching" limitation.

Continued from the previous page

burden of showing that there is no genuine issue of material fact or that or that there is an absence of evidence to support Oracle's case. *See Lujan v. Nat'l Wildlife Fed.*, 497 U.S. 871, 888-89 (1990). Nonetheless, Oracle has presented evidence showing that *all* the asserted claim limitations are anticipated by OWS 1.0. Shamos Opp. Decl. at ¶¶43-84.

[8] EpicRealm argues that Dr. Shamos did not indicate which claim construction he was applying when reaching his opinions regarding the prior art's disclosure of "dispatching." This is not true as explained in Oracle's Opposition to EpicRealm's *Daubert* Motion to Exclude the Testimony of Dr. Michael Shamos, filed herewith.

1. The Oracle WebServer User's Guide (Reference 153) Discloses "Dispatching"

a) The Oracle WebServer User's Guide Discloses "Dispatching" Under EpicRealm's Proposed Construction

The OWS Web Agent, as disclosed in Reference 153, is a "dispatcher" and performs "dispatching" under both parties' proposed constructions for those terms. Particularly, the Oracle Web Agent is a CGI program that the Web Listener executes when a request is received for a dynamic document:

> The Oracle Web Agent is a program that is invoked by the Oracle Web Listener when a request for a database procedure is received. It handles the details of making a connection to the Oracle7 Server. A Web Agent will connect to a single Oracle7 Server, using a specific database username and password which are specified as part of a Web Agent service. ***To connect to different servers, or different schemas on the same server, multiple Web Agent services may be configured on a single Oracle WebServer.*** This allows for great flexibility in the creation of applications that unify data from several different servers, while controlling exactly what information a Web client can access.

A96 at ORCL000023 (emphasis added). Accordingly, the Web Agent can connect to one of a variety of different database servers in order to dispatch a dynamic Web page request to the server for processing. The Web Agent dispatches requests according to the following steps: (1) the Web Agent receives a URL request (a dynamic Web page request) and examines it to determine the appropriate Web Agent service to use; (2) the Web Agent uses the service name to look up dynamic information in a configuration file (owa.cfg), including the identity and location of the appropriate database server to connect to; and (3) the Web Agent connects to the appropriate server and passes on the request to the appropriate server software (e.g., a PL/SQL procedure) for processing of the request. *Id.* at ORCL000115. These steps constitute dispatching a request under epicRealm's proposed construction because the Oracle dispatcher (the OWA Agent) analyzes a request (the URL) to make an informed selection of which page server (which database server) should process the request based on dynamic information

(information obtained from the owa.cfg file) maintained about page servers (the database servers), the dynamic information indicating which page server can more efficiently process the request, and sending the request to the selected page server. Shamos Opp. Decl. at ¶52.

EpicRealm makes three arguments in an effort to show that the Oracle Web Agent, as disclosed in Reference 153, is not a dispatcher and does not perform dispatching under epicRealm's proposed construction. None of epicRealm's arguments is convincing, and none certainly rises to the level of extinguishing all material issues of fact.

First, epicRealm argues that Reference 153's configuration file (owa.cfg) does not provide any information as to which page server can more efficiently process the request. EpicRealm is wrong. The configuration file provides specific information about the identity and location of the appropriate database server for processing the request. A96 at ORCL000117-118. This information identifies the "more efficient" page server because it identifies the database server that has the appropriate software (*e.g.*, PL/SQL procedure) and data needed to generate the requested Web page, as opposed to a different server that may not have all the resources required to efficiently generate the page. *See id.* at ORCL000119-120 (describing example of a Web Agent using a configuration file to identify a database server capable of invoking a "hockey_pass" PL/SQL procedure); Shamos Opp. Decl. at ¶52. A database server that is configured to process particular URL requests will certainly process such requests *more efficiently* than one that is not so configured. EpicRealm further argues that the owa.cfg file provides information about the *only* server that can process the request. But this is not necessarily true because there may exist many servers that can process a particular request and the owa.cfg file identifies the one that *should* process the request. *Id.*

Second, epicRealm argues that the information in the configuration file is not dynamic information. This is also not true. Information contained in the owa.cfg file is dynamic because

it can continuously change over time even while the WebServer software is running. A83 at pp.142:5-144:25; A96 at ORCL000173-179. For example, at one instance the configuration file can be configured to route particular requests to a particular database server using a particular username and password, and at a later instance the owa.cfg file can be modified to route those same requests to a different server using a different username and password. *See Id.* at ORCL000173-179. Although such modifications to the owa.cfg file can occur manually by a Web site administrator, the patents-in-suit do not preclude the use of manual intervention to affect how the dispatcher makes its informed selection. *See* A1 at col. 8:13-20.

Third, epicRealm argues that the Web Agent does not make an "informed selection." This is not true. As discussed above, Reference 153 discloses that OWS 1.0 may contain a variety of "different servers, or different schemas on the same server," that may all be capable of processing a particular URL request. A96 at ORCL000023. The Web Agent obtains dynamic information from the owa.cfg file, which it uses to select the appropriate page server to send the request to: "In order to connect to an Oracle7 Server, the Web Agent requires certain information, such as which server to connect to and what username and password to use. This information is stored in the owa.cfg file." *Id.* at ORCL000115. Thus, the Oracle Web Agent does choose from more than one page server.

b) The Oracle WebServer User's Guide Discloses "Dispatching" Under Oracle's Proposed Construction

Because Oracle's proposed construction is broader than and subsumes epicRealm's narrower proposed construction for "dispatching," any prior art references that discloses "dispatching" under epicRealm's proposal must also meet Oracle's proposal. Shamos Opp. Decl. at ¶52. Oracle's proposed construction is more easily met because it does not require the dispatcher to rely on *dynamic* information nor does it require the dispatcher to select the "more

efficient" page server.

Nonetheless, the Oracle Web Agent of Reference 153 does analyze a request to make an informed selection of which page server should process the request for all the reasons stated in the previous section. Again, OWS 1.0 may contain a variety of "different servers, or different schemas on the same server," that may all be capable of processing a particular URL request. A96 at ORCL000023. Accordingly, epicRealm's statement that "there is only one application on an Oracle7 Server that can process a given request" is wrong. ER Br. at p. 24. The Web Agent obtains dynamic information from the owa.cfg file, which it uses to select the appropriate page server to which to send the request: "In order to connect to an Oracle7 Server, the Web Agent requires certain information, such as which server to connect to and what username and password to use. This information is stored in the owa.cfg file." A96 at ORCL000115. Thus, the Oracle Web Agent does choose from more than one page server.

As to epicRealm's second and third arguments, Oracle did ***not*** admit that making an "informed selection" requires "dynamic information about the page servers sufficient to . . .." as alleged by epicRealm. ER Br. at p. 25. EpicRealm cites to the deposition transcript of Dr. Paul Clark as if he made such an admission, but Dr. Clark's testimony makes clear that he was referencing epicRealm's proposed construction and not Oracle's when making the statement that epicRealm quotes. *Id.* (*see* evidence cited). EpicRealm mischaracterizes Dr. Clark's testimony.

For all these reasons, Reference 153 discloses an Oracle Web Agent that is a "dispatcher" and performs "dispatching" under both epicRealm's and Oracle's proposed constructions.

2. The "Oracle WebServer: A Technical Discussion" Document (Reference 503) Discloses Dispatching Under Both Parties' Proposed Constructions

Reference 503 is a technical document authored by Katrina Montinola that Oracle distributed at a trade show in September 1995. Montinola Opp. Decl. at ¶7. As admitted by

epicRealm, Reference 503 discloses many of the same features of OWS 1.0 which Reference 153 discloses. ER Br. at p. 25. As discussed above, not all these features were before the USPTO during the initial examination of the '554 patent application. Reference 153 discloses the same Oracle Web Agent dispatching functionality that is discussed in the previous section. A41 at ORCL01042969-971. Accordingly, Reference 153 discloses "dispatching."

3. The SQL*Net References Discloses Dispatching Under Both Parties' Proposed Constructions

The SQL*Net References provide evidence of how the OWS 1.0 prior art system operated when it used SQL*Net to dispatch requests to page servers. *See* Shamos Opp. Decl. at ¶¶52-53. EpicRealm misstates Oracle's dispatching argument with respect to SQL*Net. First, as discussed above, Oracle asserts that OWS 1.0 can perform dispatching to local databases using the Web Agent *without* SQL*Net. Additionally, the Web Agent can perform dispatching to remote databases *with* the use of SQL*Net. In other words, when SQL*Net is being used, the Oracle Web Agent in combination with SQL*Net perform dispatching. Dr. Shamos disclosed this theory in his report and in his deposition. A86 at ¶110; A83 at pp. 263:2-9.

As discussed below, OWS 1.0 did in fact use SQL*Net to perform "dispatching" according to both parties' proposed constructions.

a) The SQL*Net 2.2 Reference Discloses "Dispatching"

As discussed above in Section V.B.1., *supra*, the Oracle Web Agent examines a URL request to make informed selection of which database server should process the request based on dynamic information (contained in a owa.cfg file) maintained about page servers. When the Web Agent reads the configuration file, the file can inform the Web Agent that it needs to connect to a remote database server using SQL*Net. A96 at ORCL000115-116. The Web Agent then uses SQL*Net to examine additional dynamic information to select a particular

database server that can more efficiently process the request. Particularly, the Web Agent sends a connection request to a Network Listener associated with an Oracle7 Server. Shamos Opp. Decl. at ¶¶54-55. Using SQL*Net, the Network Listener then makes an informed selection of which particular server process (or set of processes) should process the request. *Id.* As shown in the diagram at A49 at ORCL01806664, the Listener can select one of three types of servers: (a) a dedicated server; (b) a multi-threaded server; or (c) a prespawned dedicated server. Shamos Opp. Decl. at ¶¶54-55.

If the Web Agent, for example, directs the Listener to select a multi-threaded server, the dynamic Web page request is dispatched to one of several "Shared Dispatchers" for further processing. *Id.* The Listener maintains dynamic information about the load on the various Shared Dispatchers and selects the Dispatcher that can more efficiently process the request (*i.e.*, the Dispatcher that is the least loaded). *Id.* This Dispatcher is then responsible for handing off the request to a shared server that generates the dynamic Web page. *Id.*

In summary, the Web Agent uses SQL*Net to examine a request to make an informed selection of which page server (*e.g.*, 3b in Figure 2-3 at ORCL01806664 [A49]) should process the request based on dynamic information maintained about page servers (*e.g.*, information contained in the owa.cfg file and information about which Shared Dispatcher is least loaded). The dynamic information indicates which page server can more efficiently process the request because, for example, it indicates which Shared Dispatcher process is the least loaded. Naturally, a Shared Dispatcher that is the least loaded can more efficiently process request. Finally, the Web Agent sends the request to the selected page server. Accordingly, the Oracle Web Agent combined with SQL*Net, perform dispatching under epicRealm's proposed construction for the term.

EpicRealm surprisingly now disputes that the use of SQL*Net performs dispatching after

they admitted during discovery that similar functionality infringed the patents-in-suit. In epicRealm's Supplemental Response to Interrogatory No. 3, dated November 30, 2007, which called for epicRealm's infringement contentions, epicRealm alleged that Oracle's database products infringed the asserted claims because:

> a dynamic Web page generation request is routed from the Web server (Oracle HTTP Server) to a page server (Oracle's "Dispatcher" in conjunction with Oracle's shared server process(es) such as XML DB or PL/SQL). . . . The request is intercepted at the Web server and routed to a dispatcher (a portion of Oracle's "Listener").

A59 at A59-3. Accordingly, epicRealm contended that Oracle's Shared "Dispatcher in conjunction with Oracle's shared server process(es)" was a page server and that a portion of Oracle's Network "Listener" was a dispatcher of the asserted claims. After Oracle proved in discovery that this SQL*Net functionality was in the prior art, epicRealm suddenly decided that its infringement theory was no longer viable and abandoned it for a different theory.

Now that epicRealm knows that SQL*Net threatens the validity of its patents-in-suit, it has conjured up three arguments to try to eliminate SQL*Net from the case. EpicRealm's arguments are wrong under both parties' proposed claim constructions.

First, epicRealm tries to distinguish SQL*Net on the basis that a Web page request is different than a connection request. Although this can be true, epicRealm fails to show how this prevents Oracle's SQL*Net dispatching from reading on its proposed claim construction. As discussed above, the Web Agent examines a URL request (*i.e.*, a Web Page request) to make an informed selection based on dynamic information provided by SQL*Net and sends the request to the selected page server. EpicRealm does not refute this.

Second, epicRealm argues that SQL*Net is not used to make an "informed selection" because its selection of a "least called" Shared Dispatcher ***can*** be accomplished by a round-robin algorithm. ER Br. at p. 28. EpicRealm is wrong for at least two reasons. First, epicRealm does

not explain how, if at all, a selection of a "least called" Shared Dispatcher can be accomplished by round-robin, and Dr. Finkel's cited testimony on this point is completely conclusory. EpicRealm's argument should be rejected based on that point alone. *See N.L.R.B.*, 301 F.3d at 95 (A summary judgment movant's evidence "that is essentially conclusory and lacking in specific facts is inadequate to satisfy the movant's burden.") (citation omitted). Second, the SQL*Net reference discloses that the Shared Dispatchers continuously update the Network Listener with the amount of "load" on each Shared Dispatcher, and the Listener selects the least loaded Dispatcher. A49 at ORCL01806669. Such a selection method cannot be round-robin, because a pure round-robin scheme does not take into account load information obtained from the things that are being selected. *See* A83 at pp. 173:19-174:4.

Third, epicRealm argues that SQL*Net does not disclose "dispatching" because it uses a "redirect message" and, thus, does not dispatch the request. This is wrong because epicRealm ignores the fact that the Web Agent is still part of the dispatcher and it is the Web Agent that is directly sending the request to the selected page server. Shamos Opp. Decl. at ¶52; A96 at ORCL000115-116. The Web Agent's action meets the requirement of the parties' claim constructions because it is the Web Agent that eventually "send[s] the request to the selected page server." *Id.* Nevertheless, even if the Network Listener was solely the dispatcher, the parties' claim constructions do not prevent the dispatcher from sending the request by means of another piece of software (*e.g.*, by means of a redirect message). In other words, the parties' proposed constructions for "dispatching" merely require the sending of the request, and they do not specify what is doing the sending. A83 at pp. 150:21-151:24.

For all the reasons stated above, OWS 1.0 can use SQL*Net to perform dispatching under epicRealm's proposed construction for that term. At a minimum, Oracle's arguments and evidence raise a genuine issue of material fact precluding summary judgment. *See Cooper*, 748

F.2d at 679; *Armco, Inc.*, 791 F.2d at 149-51. Because Oracle's proposed construction for "dispatching" is broader than epicRealm's, OWS 1.0's use of SQL*Net meets Oracle's claim construction as well.

b) The SQL*Net 2.3 Reference Discloses "dispatching"

Release 2.3 of SQL*Net retained the same functionality as Release 2.2, except that it added a few enhancements. Shamos Opp. Decl. at ¶56. A technical description of Oracle's load balancing can be found in the "Understanding SQL*Net, Release 2.3" manual which is excerpted at A50. Particularly, SQL*Net 2.3 added "Listener Load Balancing." *Id.* at ORCL01808646-647. Unlike Release 2.2 that only allowed the Network Listener to select a page server process on the same database, the Network Listener of Release 2.3 could monitor the load on page server processes spread across multiple databases and could instruct the Web Agent to send a request to any of the page servers. *Id.*

Release 2.3 of SQL*Net discloses dispatching under both parties' proposed construction for "dispatching" for the same reasons Release 2.2 discloses dispatching, and epicRealm's arguments are wrong for the same reasons stated above. EpicRealm also argues that a "random selection" of a listener is not "dispatching," but Oracle is not relying on this random selection to meet the "dispatching" claim limitation. SQL*Net makes its informed selection *after* a listener is randomly selected: "Once randomly selected, the listeners behave as they have in previous releases, distributing connection requests to the dispatchers ***on the basis of their load*** (for multi-threaded servers) or to dedicated servers." *Id.* at ORCL01808646 (emphasis added).

4. The OWS 1.0 Product Discloses "Dispatching" Under Both Parties' Proposed Constructions

The various OWS 1.0 References discussed above all describe how the OWS 1.0 product operated at the time of its reduction to practice by July 1995 and at the time of its commercial

release in October 1995. Montinola Opp. Decl. at ¶¶6-9. Because, as demonstrated above, the above prior art references all disclose "dispatching" according to both parties' proposed constructions for that term, the OWS 1.0 product must also disclose dispatching. OWS 1.0 discloses each and every limitation of the asserted claims. Shamos Opp. Decl. at ¶¶47-82.

5. Oracle Can Combine Reference 153 with the SQL*Net References for Purposes of Proving Anticipation of the Asserted Claims

With respect to the OWS 1.0 References that are printed publications, Oracle only seeks to combine the Oracle WebServer 1.0 User's Guide (Reference 153) with the SQL*Net References for purposes of making an anticipation argument.

Material not explicitly contained in a single prior art document may still be considered for purposes of anticipation if that material is incorporated by reference into the document. *Advanced Display Sys. Inc. v. Kent State Univ.*, 212 F.3d 1272, 1282 (Fed. Cir. 2000) ("Incorporation by reference provides a method for integrating material from various documents into a host document--a patent or printed publication in an anticipation determination--by citing such material in a manner that makes clear that the material is effectively part of the host document as if it were explicitly contained therein.") (citations omitted). "[T]he standard of one reasonably skilled in the art should be used to determine whether the host document describes the material to be incorporated by reference with sufficient particularity." *Id.* at 1283.

Reference 153 references SQL*Net throughout and includes specific statements that incorporate by reference the SQL*Net manuals. *See, e.g.*, A96 at ORCL000078 and ORCL000177 (stating that more information about SQL*Net can be found in the "Understanding SQL*Net manual); *see also* ORCL000022. These statements specifically inform an Oracle customer that he or she may look to the SQL*Net manuals for more information about how SQL*Net works with Oracle WebServer 1.0. Montinola Opp. Decl. at ¶17. Reference 153 also

identifies where the SQL*Net references can be found because Oracle manuals are identified in Reference 153's text with italics, as are the SQL*Net manuals, and the reference provides information on how to contact Oracle in case such references were needed. A96 at ORCL000005 and ORCL000010. Additionally, it is likely that one who purchased Oracle WebServer 1.0, including the Oracle7 Server, would already be in possession of the SQL*Net manual as part of the product bundle. *See* Montinola Opp. Decl. at ¶8. Finally, one of ordinary skill in the art would understand that Reference 153 incorporates the SQL*Net manuals by reference. *See id.* at ¶17.

EpicRealm also makes the conclusory argument that Oracle has failed to demonstrate how each limitation of the asserted claims is disclosed by the combination of the OWS 1.0 References and the SQL*Net references. ER Br. at pp. 30-31. For example, epicRealm states, ***without providing any discussion or citing any evidence***, that Oracle has not shown that such references disclose a "Web server," a "page server," "intercepting," and "releasing." Although epicRealm is wrong, epicRealm cannot meet its initial burden on summary judgment with such conclusory statements. EpicRealm also argues that this combination of references does not disclose "dispatching," but Oracle has already demonstrated above why epicRealm's position is wrong.

Even if Oracle could not combine the above references, such references still disclose how the OWS 1.0 product operated, and Oracle relies on the prior publication, sale and public use of the OWS 1.0 product to prove anticipation of the asserted claims.

C. Oracle WebServer 2.0, as Disclosed in the Oracle References, Anticipates the Asserted Claims Under 35 U.S.C. §§ 102(a) and (g)

As shown in Fig. D below, OWS 2.0 discloses all the limitations of the independent claims including, for example, a Web client (the web browser) sending a dynamic Web page

request to a Web server (the HTTP Server). Shamos Opp. Decl. at ¶¶86-99. OWS 2.0 further discloses routing the request from the Web server to a page server (a Web Request Broker Executable Engine ("WRBX") together with a WRB Service) via a dispatcher (the WRB Dispatcher). *Id.*

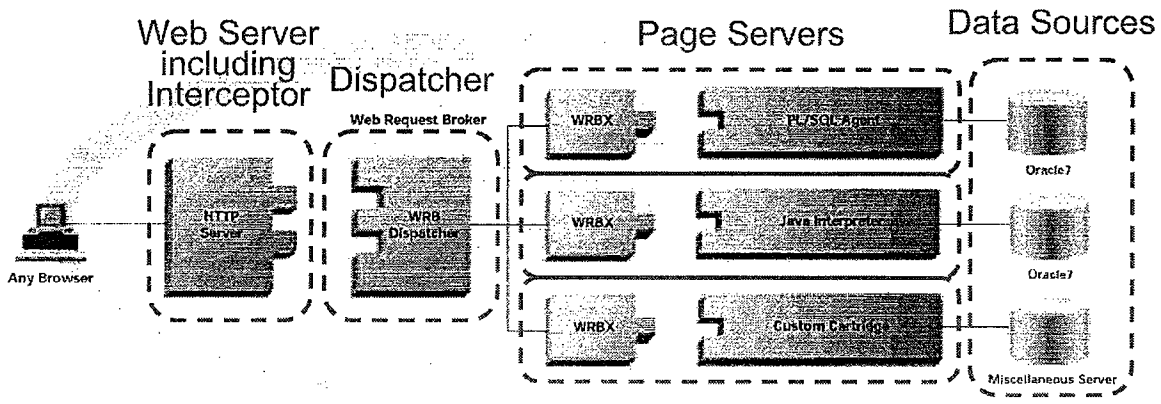


Figure D – A43 at ORCL000747 (dotted lines and top labels added)

EpicRealm incorrectly argues that Oracle has not presented evidence that a single OWS 2.0 Reference anticipates the asserted claims. Oracle's main argument with respect OWS 2.0 is that the ***product itself*** anticipates the asserted claims. Oracle has shown that the asserted claims are anticipated by the OWS 2.0 product based on the following evidence: (1) Oracle's invalidity contentions (e.g., A93 at p. 4 and Exhs. 3EE and 4EE) and (2) the expert report and testimony of Dr. Shamos (e.g., A86 at Exh. 3; Shamos Opp. Decl. at ¶¶86-124). Additionally, as discussed in Section V.A.2., *supra,* Oracle has presented evidence that the OWS 2.0 product is prior art to the patents-in-suit under 35 U.S.C. §§ 102(a) and (g) and that the relevant Oracle documentation cited in its contentions and by Dr. Shamos accurately describes how the product operated. Montinola Decl. at ¶¶10-15. Accordingly, epicRealm has not met its initial burden on summary judgment of showing an absence of evidence to support Oracle's claim.

1. The 605 Reference and the OWS 2.0 User's Guide Disclose Both "Intercepting" and "Releasing"

EpicRealm asserts that Oracle has not shown that Reference 605 (A44) and the OWS 2.0 User's Guide (A42) meet the "intercepting said request . . ." and "receiving said request and releasing . . ." limitations of the independent asserted claims. ER Br. at p. 32. But as evidenced by Dr. Shamos' report, these references do in fact disclose such limitations. In his expert report, Dr. Shamos cites to certain OWS 2.0 documents showing that the OWS 2.0 system disclosed all the limitations of the asserted claims. A86 at Exh. 3; Shamos Opp. Decl. at ¶¶86-124. The same functionality that Dr. Shamos cites in one document is equally present in other documents that describe the same operation of OWS 2.0. As shown in the chart below, Reference 605 and the OWS 2.0 User's Guide do in fact disclose the limitations that epicRealm alleges are missing:

Claim Limitations Disclosed in the OWS 2.0 References

	intercepting said request at said [Web server, second computer system or HTTP-compliant device] ('554 patent - claims 1, 9 and 11 and '335 patent at claims 2 and 16)	**said page server receiving said request and releasing said [Web server, second computer system or HTTP-compliant device] to process other requests** ('554 patent - claims 1, 9 and 11 and '335 patent at claims 2 and 16)
Reference 605	intercepting is disclosed under both parties' proposed constructions when the Oracle HTTP server "listens for static files and runs simple CGI programs, and hands off everything else to the Web Request Broker (WRB)." A44 at ORCL000764; Shamos Opp. Decl. at ¶¶92-93, 113, 120, 121 and 123.	receipt of a request and release of the Web server (or similar component) is disclosed at A44 at ORCL000766 when the WRBX instance receives and/or processes the request thereby releasing the Web server according to epicRealm's application of the parties' proposed constructions. Shamos Opp. Decl. at ¶¶89-91, 114, 120, 121 and 123.
Oracle WebServer 2.0 User's Guide	The above is similarly disclosed at, *e.g.*, A42 at ORCL01816643. Shamos Opp. Decl. at ¶¶92-93, 113, 120, 121 and 123.	The above is similarly disclosed at, *e.g.*, A42 ORCL01816652-653. Shamos Opp. Decl. at ¶¶89-91, 114, 120, 121 and 123.

2. Oracle WebServer 2.0 Discloses Dispatching Under Both Parties' Proposed Constructions

EpicRealm argues that the "Dispatcher" of OWS 2.0 is not really a dispatcher, because it

does not operate according to the extraneous limitations that epicRealm imports into the parties' proposed constructions for "dispatching." EpicRealm is incorrect and the WRB Dispatcher is in fact a dispatcher and does perform dispatching.

The operation of the OWS 2.0 Dispatcher is summarized in the Statement of Facts above. EpicRealm makes two arguments in an attempt to show that the OWS 2.0 Dispatcher is not really a dispatcher under its proposed construction. None of epicRealm's arguments, however, demonstrate that there exists no genuine issue of material fact.

First, epicRealm argues that the dynamic information indicating which WRBX page servers are connected with the requested WRB Service is not "dynamic" because such information is stored in a configuration file. Although epicRealm's configuration file argument is wrong as discussed in Section V.B.1., *supra*, the Dispatcher relies on additional dynamic information to perform dispatching.

As described in the OWS 2.0 User's Guide: "[t]he Dispatcher creates new WRBX's as needed and connects them to the appropriate WRB Services." A97 at ORCL01816653. The pool of WRBX page servers that are connected to a particular Service, therefore, is constantly changing as such processes are continuously created and destroyed. *Id.* The Dispatcher uses this dynamic information to select the appropriate set of WRBXs that can more efficiently process the request because such processes are connected to a Service that is specifically designed to handle that request. *Id.* The Dispatcher also uses additional dynamic information to determine which specific WRBX from the Service pool should process the request based on the load on that pool. *Id.* Particularly, the Dispatcher looks at the load on each WRBX and selects a free process to process the requests. Contrary to epicRealm's assertion, this load information does in fact indicate which WRBX can more efficiently process the request because, obviously, a WRBX that is free can process a request more efficiently that a process that is busy. A83 at pp. 298:1-

299:20; Shamos Opp. Decl. at ¶95. Additionally, if there is more than one free process to select from, the Dispatcher simply picks a free process based on the order in which the WRBXs were originally assigned requests. A98 at ORCL01607402.

Second, epicRealm argues that the Dispatcher is not making an "informed" selection even though it admits that the Dispatcher is relying on dynamic information including: (1) what WRB Service was called for by the request; (2) which WRBXs are connected with the requested WRB Service, and (3) load information concerning the various WRBXs. ER Br. at p. 34. EpicRealm argues that this information may only narrow down the universe of free WRBXs. But this is not true if, for example, all the WRBXs are busy, because under such circumstances the Dispatcher will spawn a new WRBX to process the request. A97 at ORCL01816653. The fact that more than one free WRBX may be available does not matter, because the Dispatcher must select one, and regardless of how it makes its final selection it will have still relied on the three pieces of dynamic information listed above. This selection must be sufficient to constitute "dispatching" because the patents-in-suit similarly fail to disclose how the dispatcher selects from identically situated page servers that all have the same load. Nonetheless, as stated above, the OWS 2.0 Dispatcher does in fact rely on additional dynamic information to select from two or more free WRBXs. It selects a free WRBX based on the order in which the WRBXs were originally assigned requests. A98 at ORCL01607402. This order can obviously change dynamically because WRBXs are constantly being created and destroyed. *Id.*; A97 at ORCL01816653.

For all of these reasons, the OWS 2.0 Dispatcher does in fact perform dispatching under epicRealm's proposed construction. Shamos Opp. Decl. at ¶¶94-95. Additionally, the Dispatcher also performs dispatching under Oracle's broader proposed construction because, as discussed above, the Dispatcher will make an informed selection of which WRBX will process the request. Oracle's proposed construction does not require that the information be dynamic nor

does it require that the selection be of a "more efficient" page server. OWS 2.0 plainly meets Oracle's proposed construction for "dispatching."

3. Oracle WebServer 2.0, as Disclosed in the Oracle References, Discloses Intercepting Under Both Parties' Proposed Constructions

EpicRealm argues that OWS 2.0 does not perform "intercepting" because, under either party's proposed construction, there is no software on the Web server that decides whether the Web server or the page server should process the request. EpicRealm is wrong. The OWS 2.0 documentation plainly discloses that the Web server distinguishes between static requests and dynamic requests, and sends certain dynamic requests to the Web Request Broker:

The HTTP server "listens for incoming HTTP requests, delivers static files and runs simple CGI programs, ***and hands off everything else to the Web Request Broker (WRB)***." A44 at ORCL000764 (emphasis added). Similarly:

> When the Web Listener receives a URL, it determines whether the request requires the use of a Service to be accessed through the Web Request Broker (WRB) . . . or whether access to the file system of the machine on which the LIStner resides is sufficient. If WRB access is required, the Listener passes the request to WRB Dispatcher for processing.

A97 at ORCL01816643. Accordingly, under both parties' proposed constructions for "intercepting," the OWS 2.0 HTTP server machine plainly includes software that will intercept requests for the purpose of routing them to the Dispatcher. Shamos Opp. Decl. at ¶¶92-93. EpicRealm's quote from an OWS 2.0 reference stating that the "HTTP engine does not make any attempt at interpreting the request" is taken out of context. This simply means that the HTTP engine does not attempt to process requests that are intended for the WRB; those requests are diverted to the WRB. A83 at pp. 295:24-297:4.

Furthermore, epicRealm argues that OWS 2.0 does not perform intercepting because it does not implement the "typical HTTP API." EpicRealm incorrectly infers that because an Oracle document states that OWS 2.0 does not implement an HTTP API, it must not perform

intercepting. Even if OWS 2.0 did not implement a typical HTTP API, epicRealm has not proven that OWS 2.0 performs intercepting by some other means. Indeed, OWS 2.0 does perform intercepting as discussed above.

In sum, epicRealm has not met its burden on summary judgment of showing that there is an absence of evidence in Oracle's case or that there is no genuine issue of material fact. To the contrary, the evidence shows that that the Oracle WebServer products and documentation anticipate the asserted claims.

VI. CONCLUSION

For the reasons stated above, the Court should deny epicRealm's Motion for Summary Judgment.

MORRIS, NICHOLS, ARSHT & TUNNELL LLP

/s/ James W. Parrett, Jr.

Mary B. Graham (#2256)
James W. Parrett, Jr. (#4292)
1201 N. Market Street
P.O. Box 1347
Wilmington, DE 19899-1347
(302) 658-9200
mgraham@mnat.com
jparrett@mnat.com
Attorneys for Oracle Corporation and Oracle U.S.A., Inc.

OF COUNSEL:

James G. Gilliland
Theodore T. Herhold
Joseph A. Greco
Robert J. Artuz
Eric M. Hutchins
TOWNSEND AND TOWNSEND AND CREW LLP
379 Lytton Avenue
Palo Alto, CA 94301
(650) 326-2400

Dorian Daley
Peggy E. Bruggman
Matthew M. Sarboraria
ORACLE CORPORATION
ORACLE U.S.A., INC.
500 Oracle Parkway
Redwood Shores, CA 94065

August 21, 2008

CERTIFICATE OF SERVICE

I hereby certify that on August 28, 2008, I caused the foregoing to be electronically filed with the Clerk of the Court using CM/ECF which will send electronic notification of such filing to the following:

Richard L. Horwitz
David Ellis Moore
POTTER ANDERSON & CORROON, LLP

Additionally, I hereby certify that true and correct copies of the foregoing were caused to be served on August 28, 2008 upon the following individuals in the manner indicated:

BY E-MAIL

Richard L. Horwitz
David Ellis Moore
POTTER ANDERSON & CORROON, LLP
1313 N. Market St., Hercules Plaza, 6th Floor
P.O. Box 951
Wilmington, DE 19899-0951

rhorwitz@potteranderson.com
dmoore@potteranderson.com

BY E-MAIL

George S. Bosy, Esquire
JENNER & BLOCK
330 N. Wabash Avenue
Chicago, IL 60611-7603

gbosy@jenner.com

/s/ James W. Parrett, Jr.

James W. Parrett, Jr. (#4292)