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.	

DECLARATION OF DR. DAVID FINKEL IN SUPPORT OF DEFENDANT'S CLAIM CONSTRUCTION BRIEF, MOTION FOR PARTIAL SUMMARY JUDGMENT OF LITERAL INFRINGEMENT AND 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
876826 / 31393 / Oracle

Public Version Dated: August 7, 2008

I, David Finkel, declare as follows:

1. I am currently the Co-Director of the Computer and Communications Networks program at Worcester Polytechnic Institute. I have been a professor of Computer Science since 1991. I submit this declaration in support of Defendant's Claim Construction Brief, Motion For Partial Summary Judgment Of Literal Infringement And Motion For Partial Summary Judgment That The Oracle References Do Not Anticipate.

2. I have been involved in studying, evaluating, and teaching about computer networks, network software, and the World Wide Web for over 15 years, and have authored numerous articles in the field. Additionally, I have taught undergraduate and graduate courses in the areas of computer networks and computer communications since 1989. Furthermore, I have taught several courses that specifically focus on Internet technology and the World Wide Web.

3. The patents in suit generally relate to a system and method for managing a request for dynamic Web page generation. In particular, the system and method described in the patents in suit describe a manner in which dynamic Web pages can be quickly and reliably generated for Web patrons visiting a particular Web site.

4. At the time the patents in suit were filed, April 1996, the use of "static" Web pages was prevalent. Typically, a static Web page might include some text and some graphics, but does not change over time. Static Web pages are usually stored on a Web server. 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.

5. As the Web developed, some Web sites began to provide dynamic Web pages, *i.e.*, Web pages that could be generated based on the specific request of the Web client. Dynamic Web pages, as opposed to static Web pages, could contain information that could

change, like the time of day, weather information, or a stock quote. 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. As the number of dynamic Web page requests increased, many issues arose, including slower response time, the failure to provide the requested Web content, or the complete failure ("crashing") of the Web server. 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. This approach, adding computer hardware, was costly and cumbersome.

6. The term "dynamic information" refers to information that can only be established during the execution of a program. Dynamic information does not refer to information that can be established prior to the execution of the program. Put another way, dynamic information is of a nature that can only be determined during runtime execution.

7. The patents in suit list "scalability" as one of the advantages of the invention. The term "scalability" refers to the ability to increase the processing capability of a Web site, for example, the ability to add page servers as needed.

8. The term "HTTP" refers to Hypertext Transport Protocol or Hypertext Transfer Protocol. The terms "Hypertext Transport Protocol" and "Hypertext Transfer Protocol" refer to the same thing.

9. The phrase "releasing the Web server to process other requests..." means that the Web server is freed to perform other tasks when the request is routed from the Web server to the page server.

10. The term "connection cache" simply refers to maintaining Cache of connections to a database.

11. "Weighed available capacity" used by Oracle Web Cache to route "cache misses" is dynamic information maintained about the application servers. It is dynamic information because it cannot be determined until runtime.

12.

REDACTED

A "work request" can include a dynamic Web page request.

13.

REDACTED

14. EpicRealm contends that "Web page" means "Web content displayable through a Web browser," and "request" means "a message that asks for a Web page." Oracle contends that "Web page" means "a file containing embedded commands in a Web formatting language such as HTML, capable of being displayed on a Web browser" and that "request" means "a message that asks for a Web page specified by a URL." Under both parties' constructions, the Accused Oracle Application Server Products route dynamic Web page generation requests. (Infringement Motion, Exh. 19, p. 2-2 to 2-5.)

15. EpicRealm contends that "Web server" means "software, or a machine having software, that receives Web page requests and returns Web pages in response to the requests." Oracle contends that "Web server" means "HTTP-compliant server software, or a machine running such software, that receives Web page requests and returns Web pages in response to the requests." Under either party's construction, the Accused Oracle Application Server Products meet the "Web server" because the Oracle HTTP Server is HTTP-compliant server software that receives Web page requests and returns Web pages in response to the requests. (Infringement Motion, Exh. 19, p. 2-3 to 2-5.)

16. EpicRealm contends that "page server" means "page-generating software that generates a dynamic Web page." Oracle contends that "page server" means "page generating software, running on a processor separate from that of the Web server, that generates dynamic Web pages." The "page server" in the Accused Oracle Application Server Products is OC4J because it includes page-generating software that generates dynamic Web pages. (Infringement Motion, Exh. 19, p. 2-6 to 2-9.) In addition, under Oracle's construction, OC4J can run on a processor separate from that of the Oracle HTTP Server. The "page server" (OC4J) receives the request because, for example, OC4J can generate a response to the request. (Infringement Motion, Exh. 19, p. 2-6 to 2-9.)

17. Regarding Claim 11 of U.S. Patent No. 5,894,554, when the "Web server" receives a request, it is assigned the task, duty or obligation to process the request (*e.g.*, generate a Web page). Upon identifying that the request can be satisfied by a "page server" (OC4J), the request is routed from the "Web server" to a "page server." When the "page server" receives the request processes the request, thus freeing the "Web server" of the task, duty or obligation to satisfy the request.

18. As a non-technical analogy to the concept of paragraph 17, when an employer accepts work from a client, the employer is assigned the task, duty or obligation to do the work for that client. However, the employer is freed to do work for other clients when the employer assigns the work to an employee.

19. Also regarding Claim 11 of U.S. Patent No. 5,894,554, the "Web server" is freed by the "page server" acknowledging receipt of a request. The acknowledgement message indicates to the sender that its request has been transmitted successfully; the receipt of this acknowledgment allows the "Web server" to release resources associated with this request.

20. Continuing the non-technical analogy from paragraph 18, but for the employee's acceptance of an assignment, the employer would be obligated to do the work or to assign it to another employee.

21. Oracle HTTP Server can handle some requests on its own. For example, the Web server can handle requests for static content on its own by retrieving the content from its file system. (Infringement Motion, Exh. 11, p. 3-5.) Other requests, such as dynamic Web page generation requests, are intercepted at the Oracle HTTP Server and, instead, routed to a dispatcher for processing by a page server, *e.g.* OC4J. (Infringement Motion, Exh. 19, p. 2-2 to 2-15.) Therefore, the handling of a dynamic Web page generation request can be intercepted at Oracle HTTP Server. Similarly, when Oracle HTTP Server receives a dynamic Web page generation request, the request can be diverted before Oracle HTTP Server can process the request. In such cases, the request can be sent, instead, to a page server, *e.g.* OC4J, for processing.

22. As stated above, OC4J can process a request. (Infringement Motion, Exh. 30, Nelson Dep. 124:2-9.) The processing by OC4J is performed while the Oracle HTTP Server

concurrently processes other requests. Where the Oracle HTTP Server and the OC4J are running on the same machine, their processing is performed concurrently because both Oracle HTTP Server and the OC4J are in a processor queue at the same time. Where the Oracle HTTP Server and OC4J are running on different machines, their processing is performed concurrently when Oracle HTTP Server and OC4J are in processor queues at the same time.

23. Under Oracle's construction, "data sources" are "computerized repositories from which data may be retrieved electronically." EpicRealm believes that no construction is necessary for data sources." Under Oracle's construction or the ordinary meaning, the Accused Oracle Application Server products can retrieve data from one or more "data sources" including Oracle Database. In addition, the Accused Oracle Application Server Products can dynamically generate a "Web page" using a page server (OC4J), for example, using a Java servlet, JSP or EJB. (Infringement Motion, Exh. 9, ORCL00338431.) The "Web page" can include data retrieved from one or more "data sources." For example, a Java servlet "receives a request from a client, dynamically generates the response (possibly querying databases to fulfill the request), and then send [sic] the response containing an HTML or XML document to the client." (Infringement Motion, Exh. 20, ORCL00010749.)

24. The Accused Oracle Database Products can use any commercially available Web server including Oracle HTTP Server. (*See, e.g.*, Oracle HTTP Server, mod_plsql User's Guide, 10g Release 2 (10.2), B14337-01 ("mod_plsql is an Oracle HTTP Server plug-in that communicates with the database.").)

25. A RAC instance in the Accused Oracle Database Products meets either party's construction of "page server." A RAC instance can generate a dynamic Web page in at least three ways: (1) operate XML DB (Infringement Motion, Exh. 12, p. 1-17, 3-70 to 3-71);

(2) execute a PL/SQL Web Application (Figure 11-1 from Infringement Motion, Exh. 8); (3) execute a PL/SQL Server Page (Infringement Motion, Exh. 8, p. 12-1). Further, a RAC instance can be installed on a separate machine than the "Web Server" (*e.g.*, Oracle HTTP Server). In that case, the RAC instance would necessarily run on a separate processor than the "Web server."

26. The information used by Oracle Call Interface to select a session from its session pool based on tagging can come from examining a Web page request.

27. Security attributes and privileges can come from a dynamic Web page request.

28. The information used by JDBC to "identify connections that are authenticated with the proper user credentials, user name and password" and to "identify connections that have the necessary striping" can come from a dynamic Web page request.

29. The processing by a RAC instance can occur while Oracle HTTP Server concurrently processes other requests. Where the Oracle HTTP Server and a RAC instance are running on the same machine, their processing is performed concurrently when both the RAC instance and Oracle HTTP Server are in a processor queue at the same time. Where the Oracle HTTP Server and OC4J are running on different machines, their processing is performed concurrently when the Oracle HTTP Server and the RAC instance are in processor queues at the same time.

30. In the Accused Oracle Database Products, the common pool of disks that can be accessed by the RAC instances meets the ordinary meaning and Oracle's construction of "data sources." These common pools of disks are computerized repositories from which data may be retrieved electronically. As explained above, the RAC instances can generate a dynamic

"Web page" in at least three ways. In each of these cases, the RAC instance can retrieve data from one or more "data sources" and include that data in the dynamically generated "Web page."

31. Under both parties' constructions, the Accused Oracle Web Cache Products route dynamic Web page generation requests. (*See, e.g.,* Infringement Motion, Exh. 11, ORCL00473220-2.) In addition, the Accused Oracle Web Cache Products include a "Web server" under either party's construction because Oracle Web Cache is HTTP-compliant server software that receives Web page requests and returns Web pages in response. (*See, e.g.,* Infringement Motion, Exh. 9, ORCL00513911-913.) The "page server" in the Accused Oracle Web Cache Products can be Oracle Application Server because it includes page-generating software that generates dynamic Web pages, *e.g.*, OC4J, Perl, and CGI. (Infringement Motion, Exh. 9, ORCL00338431.) Oracle Application Server can run on a processor separate from that of Oracle Web Cache because, for example, it can run on a separate machine than Oracle Web Cache. When a request is routed to Oracle Application Server, it receives the request because, as explained above, it generates a response to the request.

32. Oracle Web Cache can handle some requests on its own. For example, the Oracle Web Cache can return pre-existing content on its own by retrieving the content from its cache. (Infringement Motion, Exh. 11, pp. 1-1 to 1-2.) Other requests, such as dynamic Web page generation requests that are "cache misses" are intercepted at Oracle Web Cache and instead routed to a dispatcher for processing by a "page server" (*e.g.*, Oracle Application Sever). (Infringement Motion, Exh. 7, Clark Rpt. 10.) Thus, the handling of a dynamic Web page generation request can be intercepted at Oracle Web Cache. The Accused Oracle Web Cache Products receive a dynamic Web page generation request at the machine running Oracle Web Cache and divert the request before the Oracle Web Cache executable can process the request

because the request is instead sent to a page server (*e.g.*, an Oracle Application Server) for processing. (Infringement Motion, Exh. 7, Clark Rpt. 10.)

33. The weighted available capacity used by Oracle Web Cache to route stateless Web cache misses is dynamic information because it cannot be generated prior to runtime.

34. As stated above, Oracle Application Server can process a request, *e.g.*, generate a dynamic Web page in response to the request. The processing by Oracle Application Server can be concurrent with the processing of other requests by Oracle Web Cache because, for example, Oracle Web Cache can process one request, *e.g.*, return a cached page, while an Oracle Application Server can process another request.

35. Oracle Application Server can generate a dynamic Web page that includes date retrieved from one or more data sources for example, by using CGI or Perl.

36. Neither SQL*Net 2.2 nor SQL*Net 2.3 could generate a dynamic Web page by themselves. (*See* Motion for Partial Summary Judgment That the Oracle References Do Not Anticipate ("Anticipation Motion"), Exhs. 19 and 21.)

37. The configuration file used by the Web Agent in Oracle 1.0 does not provide any information on which application on an Oracle7 Server can more efficiently process a dynamic Web page request. (*See* Anticipation Motion, Exh. 3, Shamos Rpt. ¶548; Anticipation Motion, Exh. 5, ORCL00941822-26.)

38. Neither SQL*Net Reference discloses "intercepting," "page server," "concurrent processing," and/or "dispatching." (*See* Anticipation Motion, Exhs. 19 and 21.)

39. A request for a database connection attempts to establish a connection between a client and a server; the response to a request for a database connection will not be a

dynamic Web page. 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.

40. One way of implementing the selection of the "least called" server is a round-robin algorithm.

41. When a client is connecting to a multi-threaded database using SQL*Net 2.2 or SQL*Net 2.3, the request for a database connection sent from the client to the database is a wholly different request for a connection than the request originally received by SQL*Net.

42. I am over the age of eighteen, and, if called to do so, I am prepared and competent to testify on the matters presented in this declaration.

I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct. Executed on July 31, 2008, in Boston, Massachusetts.



Dr. David Finkel

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