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.	)	

SECOND DECLARATION OF DR. DAVID FINKEL IN SUPPORT OF EPICREALM'S OPPOSITIONS TO ORACLE'S MOTIONS FOR SUMMARY JUDGMENT OF NONINFRINGEMENT AND INVALIDITY

VOLUME I of V - (PART A)

EXHIBITS 1 to 4

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: August 21, 2008
Public Version Dated: August 29, 2008
880482 / 31393 / Oracle

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

ORACLE CORPORATION and ORACLE U.S.A. INC.,	)	
Plaintiffs,	)	C.A. No. 06-cv-414 (SLR)
v.	)	
EPICREALM LICENSING, LP,	)	
Defendant.	)	

SECOND DECLARATION OF DR. DAVID FINKEL IN SUPPORT OF EPICREALM'S OPPOSITIONS TO ORACLE'S MOTIONS FOR SUMMARY JUDGMENT OF NONINFRINGEMENT AND INVALIDITY

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 EpicRealm's Oppositions to Oracle's Motions For Summary Judgment of Noninfringement and Invalidity.

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 Accused Oracle Application Server Products literally infringe Claim 1 of the '554 Patent under both parties' constructions. (*See* 2d Finkel Decl. Ex. 1 at 1-20.)

4. The Accused Oracle Application Server Products literally meet the "intercepting" limitation under either party's construction. (*See* 2d Finkel Decl. Exs. 1, 2.)

5. The "Web server" in all of the Accused Oracle Products processes some, but not all requests.

6. All software always goes through its "same iterative process," *i.e.*, it always follows its sequence of instructions. The software code does not change.

7. In Leaf, all requests are immediately forwarded on from the so-called "Web server" to transaction gateways; the "Web server" in Leaf could not process requests itself. Thus, a request could not be "intercepted" at the "Web server" in Leaf because there is no determination of whether the "Web server" or a "page server" should process the request—the

"Web server" in Leaf would never process a request itself. Therefore, Leaf did not disclose "intercepting."

8. The Accused Oracle Application Server Products literally meet the "releasing" limitation under either party's construction because the "Web server" (OHS) is released to process "other requests." (*See* 2d Finkel Decl. Exs. 1, 2.)

Redacted

Redacted

12. The Accused Oracle Application Server Products literally meet the "dispatcher" limitation under both its ordinary meaning and Oracle's construction. (*See* 2d Finkel Decl. Ex. 1, 2.)

13. Mod_oc4j meets Oracle's construction of "dispatcher" because it is a software program that determines which page server should process a dynamic Web page generation request.

14. Source code is written in modular fashion. For example, within one piece of software there can be one section of code that performs the "Web server" functionality and another section of code that performs the "dispatcher" functionality. Therefore, a request is routed from the "Web server" to the "dispatcher" when the request is routed from the code that performs the "Web server" functionality to the code that performs the "dispatcher" functionality.

15. The "Database Connector" is a Dynamically Linked Library (".dll") that runs in the same executable as the IIS 1.0 "Web server."

16. The Accused Oracle Database Products literally infringe claim 1 of the '554 Patent under both parties' constructions. (*See* 2d Finkel Decl. Ex. 3 at 1-21.)

Redacted

Redacted

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

23. The Accused Oracle Web Cache Products literally infringe Claim 1 of the '554 Patent under both parties' constructions. (*See* 2d Finkel Decl. Ex. 5 at 1-19.)

24. The "Web server" (Web Cache) processes some but not all requests.

25. Web Cache is released to process other requests under either party's construction of "releasing." When Web Cache routes a request to a "page server," it is released to process

other requests under both parties' constructions in the same two ways—the same two ways that the "Web server" of the Accused Oracle Application Server Products are released.

Redacted

28. Merely having a "page server" process the request does not necessarily mean that the "releasing" limitation is met. For example, a system could be designed to have the "Web server" process a request in parallel with a "page server," *e.g.*, to help ensure a prompt response to a Web client even if the "page server" subsequently became slow or had a problem.

Redacted

31. The Accused Oracle Products also literally infringe the asserted claims that depend from Claim 1 of the '554 Patent. (*See* 2d Finkel Decl. Exs. 1, 3, 5.)

32. The Accused Oracle Products also literally infringe Claims 9 and 10 of the '554 Patent. (*See* 2d Finkel Decl. Exs. 1, 3, 5.)

33. The Accused Oracle Products also literally infringe Claim 11 of the '554 Patent. (*See* 2d Finkel Decl. Ex. 1, 3, 5.)

34. The Accused Oracle Products also literally infringe Claims 2 and 16 of the '335 Patent. (*See* 2d Finkel Decl. Ex. 2, 4, 6.)

35. Oracle provides various manuals to its customers instructing them how to use the Accused Oracle Products in an infringing manner.

36. The Accused Oracle Products infringe the asserted claims under the doctrine of equivalents. The Accused Oracle Products infringe because the differences between the Accused Oracle Products and the asserted claims are insubstantial from the perspective of a person of ordinary skill in the relevant art.

37. Nowhere in the specification does it discuss that "intercepting a request" requires that the Web server distinguish between dynamic and static content. Nor would one skilled in the art understand that the claims of the patents require such a distinction. Oracle Web Cache makes a determination of whether it will process a request itself (*e.g.*, satisfy the request with data stored in its cache) or it will send the request to an application server for processing.

38. The partitioned architecture of the claimed invention involves adding "page servers" which are capable of processing dynamic Web page requests. The "page servers" alleviate the processing burden placed on the "Web server" by processing some or all of the dynamic Web page requests. When a "page server" processes a request it "frees" or "releases" the "Web server" to process other requests because the "Web server" no longer has to expend resources processing that request. .

39. In the patented invention, the "dispatcher" can reside on a different machine than the "Web server" or the "dispatcher" can reside on the same machine as the "Web server." Regardless of where the "dispatcher" is located, it performs the same function. When the "dispatcher" resides on the same machine as the "Web server," some of the advantages of the invention will not be enjoyed. Namely, the Web server will not be able to offload the processing of the requests when a request is routed to the "dispatcher." The Web server, however, may still be able to offload the processing of a request when the request is routed from the "dispatcher"

residing on the Web server machine to a "page server." Thus, the purpose of the "dispatcher" is to "dispatch" requests, not to unburden the "Web server."

Redacted

41. The Oracle Web Agent does not make an informed selection of which page server should process the request. When the Oracle Web Agent determines which "page server" (Oracle7 Server) should process a request, it does not make a selection between Oracle7 Servers that are capable of processing the request. Rather, only one Oracle7 server could process the request. Oracle concedes that making an "informed ***selection***" requires, at least, choosing from more than one page server that is capable of processing the request. Thus, the Oracle Web Agent does not make an "informed selection."

42. The Oracle Web Agent also does not make a selection based on information indicating which page server can more efficiently process a request. Because only one Oracle7 server can process a given request, the information used by the Oracle Web Agent does not indicate the relative efficiencies of the Oracle7 servers. Instead, the information only indicates

which Oracle7 server to connect to and how to connect to that server. Thus, the Oracle Web Agent does not use information indicating which page server can more efficiently process a request in deciding which page sever should process a request.

43. 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.

44. SQL*Net is remote data access software. Its fundamental purpose is to allow clients and servers to access Oracle databases from remote locations. SQL*Net is not concerned with the management of dynamic Web page requests. In fact, the term "Web page" is never used in the "Understanding SQL*Net 2.2" reference. Load balancing of connection requests has little to do with metric-based load balancing of dynamic Web page requests.

45. SQL*Net routes requests for database connections which are not the same as dynamic Web page requests as recited in all of the asserted claims, which are a type of "work requests." 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.

46. 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.

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

48. Oracle 1.0 does not anticipate any of the asserted claims.

49. Claim 4 requires that the "page server" maintain a connection cache to said one or more data sources. Claim 5 requires that the processing step performed by the "page server" also includes the step of logging into one or more data sources. Oracle 1.0 does not anticipate Claims 4 and 5 of the '554 Patent for the additional reason that it cannot maintain a connection cache or log into one or more data sources. According to Oracle, the "page server" in Oracle 1.0 is the Oracle7 server, *i.e.* an Oracle database server. The data sources in Oracle 1.0 are the Oracle7 database. The Oracle7 server does not need to establish a connection to an Oracle7 database in order to retrieve data from the database because they are located on the same machine.

50. Oracle argues that Claim 5 is met because the "dispatcher" (the Oracle Web Agent) logs into the Oracle7 server. However, Claim 5 states that the "processing" step includes logging into one or more data sources and the Oracle Web Agent does not perform the "processing" step of Claim 1. Indeed, Oracle contends that the processing step is performed by the Oracle7 server, not the Oracle Web Agent.

51. Oracle 2.0 does not anticipate any of the asserted claims.

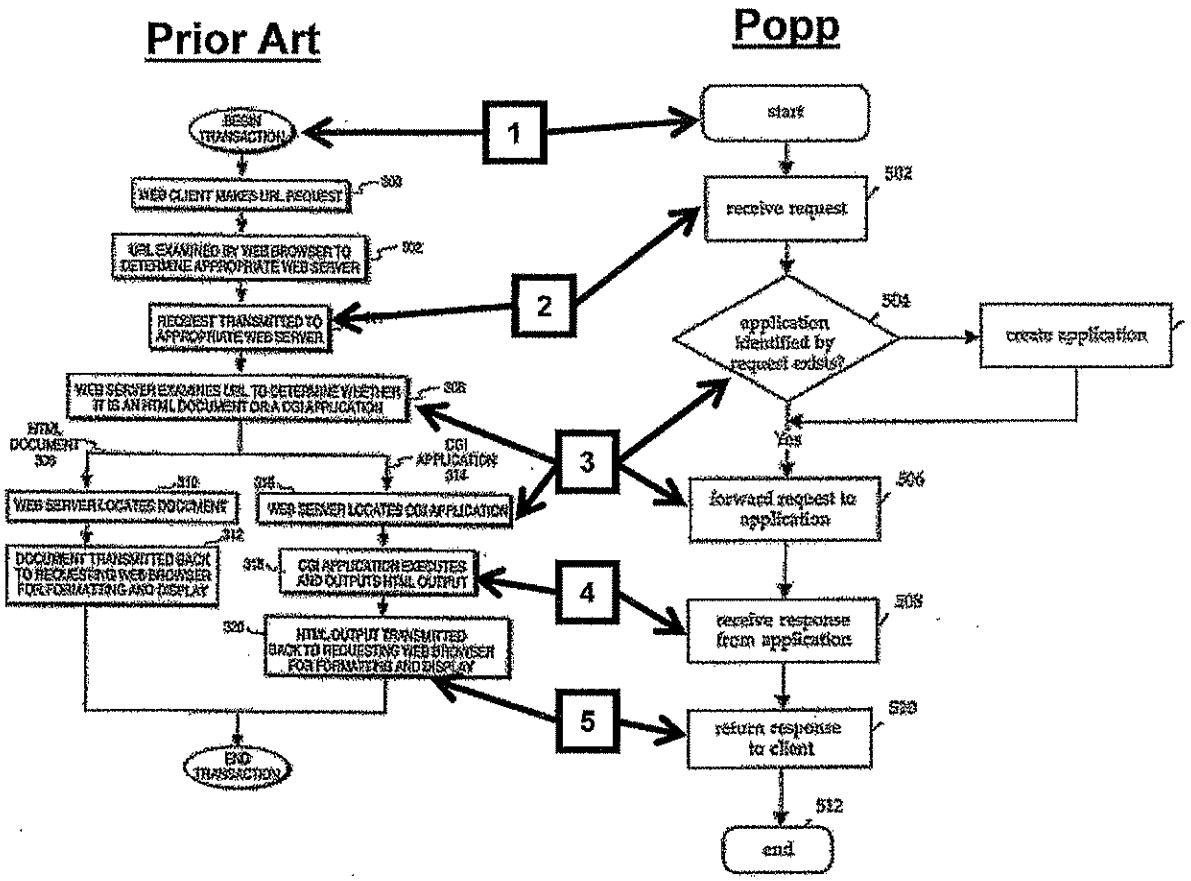
52. Oracle's proposed construction for page server is "page generating software, running on a processor separate from that of the Web server, that generates dynamic Web pages." However, Oracle's alleged "page server" in Oracle 2.0 is a WRBX that runs on the same machine (and therefore the same processor) as the Oracle 2.0 "Web server."

53. Oracle 2.0 does not anticipate Claim 4 for the additional reason that it does not disclose the "page server maintaining a connection cache to said one or more data sources." Under the ordinary meaning of "connection cache," as understood by one skilled in the art, maintaining a connection cache involved a cache of connections to a database that can be

checked out and reused by numerous processes. When a PL/SQL Agent stays connected to a database, there is neither a cache of connections nor can that connection be shared among various processes. Thus, it fails to meet the ordinary meaning of "connection cache."

54. Popp does not anticipate any of the asserted claims.

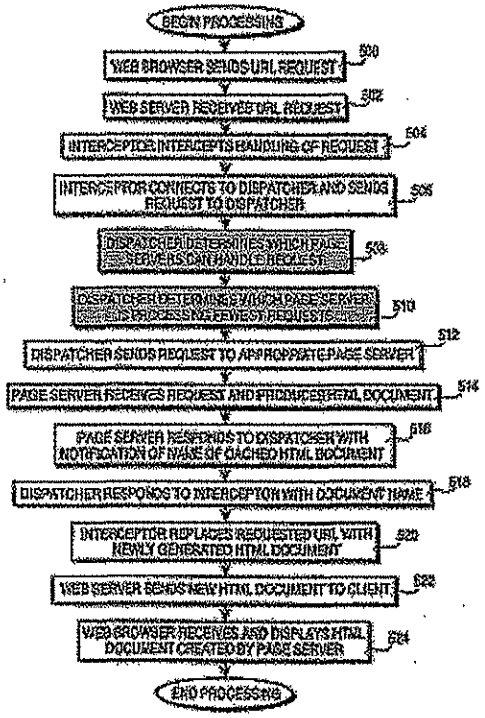
55. Below on the left is a flow chart describing how a request is processed in the prior art Web server disclosed by the '554 Patent and on the right is a flow chart from Popp describing how it processes a request:



nearly identical manner as the prior art disclosed by the patents in suit and considered by the

examiner. Specifically, Popp only determined if the application identified by a request existed, and if it did, forwarded the request to that application.

56. By comparison, in the flow diagram below depicting an embodiment of how the invention of the patents in suit processed a request, the "page server" is selected based on a two-step process (Steps 508 and 510 highlighted in gray below): (1) determine "which page servers can handle request;" and (2) determine "which page server is processing fewest requests" (*i.e.*, which "page server" can more efficiently process the request).



57. Popp's approach to Web site architecture is very similar to Oracle 1.0. In both cases, the primary concern was routing Web page requests to an application that could process the request. Neither Popp, nor Oracle 1.0, gave any consideration to determining which instance of an application "can *more efficiently* process the request." In fact, Oracle 1.0 and Popp are nearly identical to the prior art described in the '554 Patent specification.

58. By comparison, as shown in the embodiment illustrated in Figure 5 of the '554 Patent, the "page server" is selected using steps 508 ("dispatcher determines which page servers can handle request") and 510 ("dispatcher determines which page server is processing fewest requests"), *i.e.*, which "page server" can more efficiently process the request.

59. The CGI interface program in Popp (the alleged "dispatcher") does not make an "informed selection" of which page server should process a request. When the CGI interface program selects an application to process a request, it only determines which application can process a request because there is only one application that can process each request.

60. The CGI interface also does not make a selection based on information indicating which "page server" can more efficiently process a request. Because only one application can process a given request, the information used by the CGI interface does not indicate the relative efficiencies of the applications.

61. The information used by the CGI interface program in Popp is not "dynamic information maintained about page servers." The information relating to which application can process a request and on which server the application is executing is static information because it can be determined before runtime execution.

62. The CGI interface also does not make an "informed selection" because it does not use any information "to make a reasonable or accurate determination as to which [page server] could more efficiently process the request."

63. Dienst does not anticipate any of the asserted claims.

64. Dienst is not concerned with determining which "page server" can more efficiently process a request. Instead, it is concerned with locating the only "page server" that can process a request.

65. Dienst does not disclose "dispatching" for many of the same reasons that Oracle 1.0 and Popp do not disclose "dispatching." All these references are focused on locating the one and only "page server" that can process a given request.

66. The alleged "dispatcher" in Dienst does not make an "*informed selection*" of which page server should process a request. In Dienst, only one "page server" (Dienst Server) contains all the works for a specific publisher. Thus, when the alleged "dispatcher" selects a Dienst Server to process a request, it does not make a selection, but rather locates the only Dienst Server that contains that publisher's works.

67. The alleged "dispatcher" in Dienst also does not make a selection based on information indicating which "page server" can more efficiently process a request. The alleged "dispatcher" only uses information on which Dienst Server is capable of responding to a given request. Because there is only one Dienst Server that can process a request, there is no information on which Dienst Server can more efficiently process that request.

68. The information used by the CGI interface program is not dynamic information maintained about "page servers" indicating which "page server" can more efficiently process a request. Each Dienst Server downloads a listing of the locations of the other Dienst Servers so that it can search the other servers for works. This information is not dynamic information maintained about "page servers" indicating which "page server" can more efficiently process a request.

69. In my opinion, a person of ordinary skill in the art would have a Bachelor's degree in computer science or equivalent, with three to five years of technical experience in software implementation, or equivalent.

70. There is no "dispatching" under either party's proposed construction because the job of the alleged "dispatcher" is limited to locating the one "page server" that can process the given request.

71. Garland does not cure the "dispatching" deficiency in Oracle's invalidity arguments. The "Dispatcher Server" in Garland does not "examine" or "analyze" the request, as required by both parties' "dispatching" constructions. The "Dispatcher Server" also does not send the request to the selected server as required by both parties' "dispatching" constructions.

72. None of the asserted claims are obvious over the combinations of references argued by Oracle.

73. Oracle has not proved by clear and convincing evidence that Garland in combination with any of Oracle's four other references would have rendered obvious to one of ordinary skill in the art any of the twelve asserted claims at the time of the respective inventions.

74. Because Dienst is concerned with providing access to a distributed digital library over the Web, it does not disclose many of the limitations added by the dependent claims of the '554 Patent, which are focused on features to further improve the performance of a Web site. Oracle argues that it would be obvious to combine Dienst with the teachings of Oracle 1.0 or Oracle 2.0 to meet these claims. In addition, none of these references discloses "dispatching." Their combination would not change that fact. Not only does Oracle fail to set forth its basis for claiming that it would be obvious to combine these references, but the secondary considerations argue against such combinations.

75. Further, it would not be obvious to combine Dienst, a reference on providing access to a distributed digital library with Oracle 1.0 or Oracle 2.0. In fact, it would be impossible to use the "page servers" of Oracle 1.0 or Oracle 2.0 to implement Dienst.

76. 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 the request.

77. 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 August 21, 2008, in Texarkana, Texas.

Dr. David Finkel

AUG-21-2008 23:04 P.17

TOTAL P.17

TABLE OF CONTENTS

Exh.	*Document*
1	Claim Chart U.S. Patent No. 5,894,554 – The Accused Oracle Application Server Products
2	Claim Chart U.S. Patent No. 6,415,335 – The Accused Oracle Application Server Products
3	Claim Chart U.S. Patent No. 5,894,554 – The Accused Oracle Database Products
4	Claim Chart U.S. Patent No. 6,415,335 – The Accused Oracle Database Products
5	Claim Chart U.S. Patent 5,894,554 – Accused Oracle Web Cache Products
6	Claim Chart U.S. Patent No. 6,415,335 – The Accused Oracle Web Cache Products
7	Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)
8	Popp et al. U.S. Patent No. 6,249,291 (195)
9	Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)
10	Oracle WebServer User's Guide Release 2.0 Production, Part No. A23646-1 and Oracle WebServer 2.0 Technical Note
11	Functional Specification for Runtime Load Balancing in OCI Session Pools, 11gR1, 12/2/05, CHID00081-00105
12	Excerpt from deposition transcript of Lakshminarayanan Chidambaran, Feb. 29, 2008
13	Excerpt from deposition transcript of Carol Colrain, April 15, 2008
14	Excerpt from deposition transcript of Hal Hildebrand, Dec. 21, 2007
15	Excerpt from deposition transcript of Matthew Mayerson, April 15, 2008
16	Excerpt from deposition transcript of Mark Nelson, Feb. 22, 2008
17	Excerpt from deposition transcript of Douglas Surber, Feb. 28, 2008
18	Design Specification for OCI Runtime Connection Load Balancing in 11g, 12/14/05, ORCL01310468-01310498
19	Functional Specification for OFCGI, OHS, iAS, 11AS, 6/2/06, ORCL01209371-01209387

20 Functional Specification for Runtime Load Balancing in OCI Session Pools, 11gR1, 12/2/05, ORCL01166752-01166776

21 Functional Specification for Runtime Load Balancing, RDBMS, 10gR2, 4/1/04, ORCL01060148-01060193

22 Functional/Design Specification for Runtime Work Request Load Balancing using RAC Service Metrics, JDBC, 10gR2, 5/1/04, ORCL01054844-01054858

23 Excerpt from Oracle Application Server Containers for J2EE Support for JavaServer Pages Developer's Guide 10g (9.0.4), B10320-01 Sept. 2003, ORCL00337734-00338101

24 Excerpt from Oracle Application Server Containers for J2EE, Support for JavaServer Pages Developer's Guide 10g Release 2 (10.1.2) B14014-02, July 2005, ORCL00444343-00444558

25 Excerpt from Oracle Application Server Web Cache Administrator's Guide 10g Release 2 (10.1.2) B14045-01, Dec. 2004, ORCL00473184-00473739

26 Excerpt from Oracle Call Interface Programmer's Guide, 11g Release 1 (11.1), B28395-02, Oct. 2007

27 Excerpt from Oracle Call Interface Programmer's Guide, 11g Release 1 (11.1), B28395-03, May 2008

28 Excerpt from Oracle Database Administrator's Guide, 10g Release 2 (10.2), B14231-02, May 2006, ORCL01969895-01970763

29 Excerpt from Oracle HTTP Server Administrator's Guide 10g (9.0.4), B10381-02, March 2004, ORCL00372061-00372362

30 Oracle Real Application Clusters 11g: An Oracle Technical White Paper, April 2007, ORCL01972320-01972333

31 Excerpt from Oracle Web Cache Administration and Deployment Guide, Release 1.0.2, A86722-01, Nov. 21, 2000, ORCL00107955-00108126

32 Excerpt from Oracle XML DB Developer's Guide 11g Release 1 (11.1), October 2007, ORCL01972631-01973585

33 Excerpt from Oracle9i Application Serer: Availability, Scalability and Manageability of J2EE and Web Cache Clusters, An Oracle White Paper, May 2002 ORCL00013356-00013398

34 Excerpt from Oracle9i Application Server Performance Guide, Release 2 (9.0.2), April 2002, A95102-02, ORCL00847081-00847294

35 Excerpt from Oracle9i Application Server, An Oracle White Paper, Sept. 2002, ORCL00010730-00010792

36 Excerpt from Oracle9i Application Server, Best Practices, Release 1 (v1.0.2.2) Sept. 2001 A95201-01, ORCL00190234-00190397

37 Oracle9i Application Server, ORCL00110683

38 Excerpt from Oracle9i JDBC Developer's Guide and Reference Release 2 (9.2) March 2002, ORCL00306997-00307580

39 Excerpt from Oracle9iAS Web Cache Administration and Deployment Guide, Release 2 (9.0.2), May 2002, Part No. A95404-02, ORCL00272342-00272887

40 Excerpt from Oracle9iAX Web Cache Administration and Deployment Guide, Release 2.0.0, Oct. 10, 2001, A90372-04, ORCL00189864-00190165

41 Presentation, Mod_oc4j Load Balancing in iAS 9.0.4, ORCL00018115-00018150

42 Excerpt from Plaintiff's Responses to Defendants' First Set of Interrogatories (1-6)

43 Tanenbaum, A.S., *Computer Networks*, Fourth Edition. Prentice-Hall PTR, Upper Saddle River, NJ, 2003.

44 Oracle WebServer User's Guide Release 1.0, A349862 ORCL00941713-00941960 (MAC003037-003284) (*Ref 153*)

45 US 6,651,108, Popp et al., MAC003907-003927 (*Ref. 194*)

46 US 6,249,291, Popp et al., MAC003928-003968 (*Ref. 195*)

47 Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2, 1996, ORCL00000782-00001035 (*Ref. 604*)

48 Oracle WebServer 2.0 Technical Note, March 1996, ORCL00000761-00000781 (*Ref. 605*)

49 Oracle and the Internet, Using Oracle in the Emerging Electronic Infrastructure of Business and Society, July 1995, ORCL00937779-00937806

50 Excerpt from Understanding SQL*Net, Release 2.3, A42484-1, 1996, ORCL01808597-01808820

51 The Rush to the Net, May 1996, ORCL01606514-01606526

52 Web Request Broker API, Ver. 0.8, Feb. 12, 1996, ORCL01606852-01606867

53 Excerpt from deposition transcript of Katrina Montinola, Feb. 19, 2008

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

CERTIFICATE OF SERVICE

I, David E. Moore, hereby certify that on August 29, 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 29, 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

Exhibit 1

THIS EXHIBIT HAS BEEN REDACTED IN ITS ENTIRETY

Exhibit 2

THIS EXHIBIT HAS BEEN REDACTED IN ITS ENTIRETY

Exhibit 3

THIS EXHIBIT HAS BEEN REDACTED IN ITS ENTIRETY

Exhibit 4

THIS EXHIBIT HAS BEEN REDACTED IN ITS ENTIRETY