IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

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

EXHIBITS 5 to 10

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,	)	
v.	)	C.A. No. 06-cv-414 (SLR)
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.

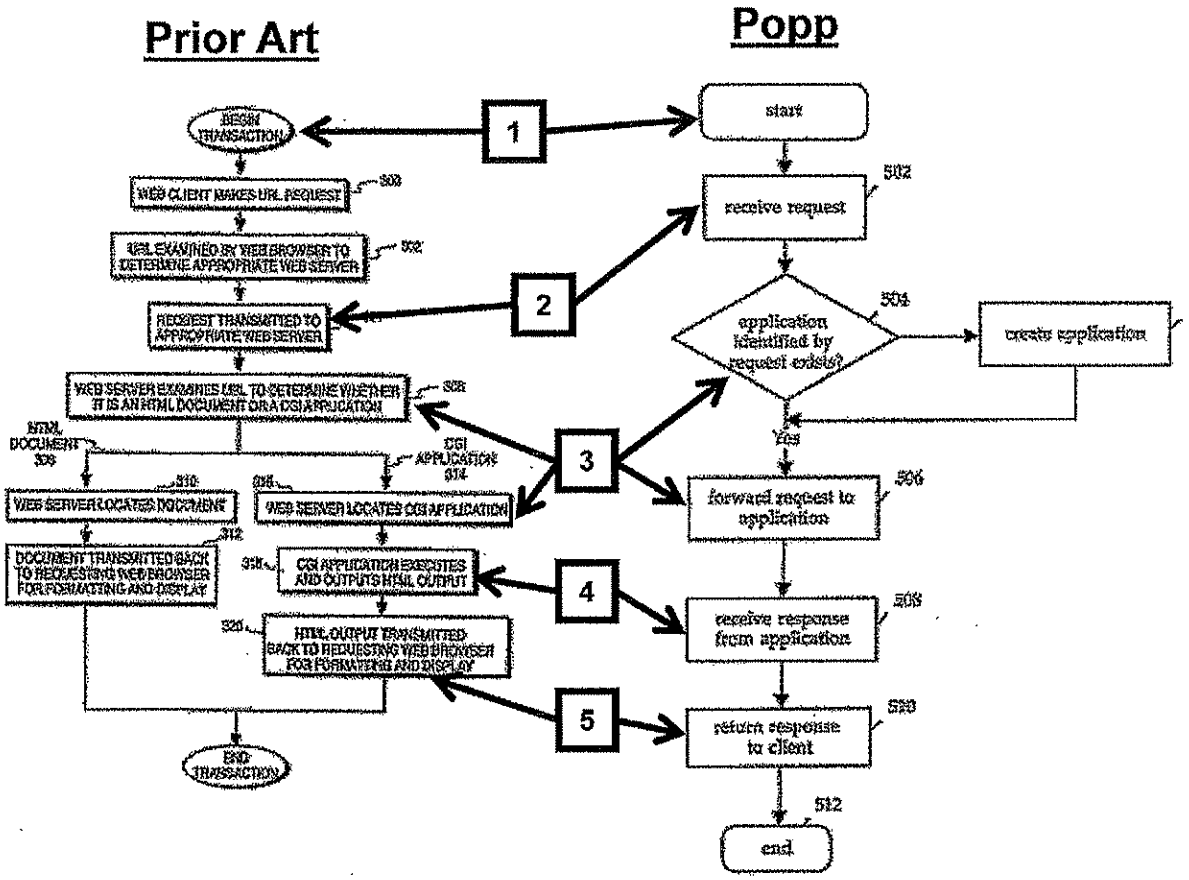
52. Oracle's proposed construction for page server is "page generating software, running on a processor separate from that of the Web server, that generates dynamic Web pages." However, Oracle's alleged "page server" in Oracle 2.0 is a WRBX that runs on the same machine (and therefore the same processor)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

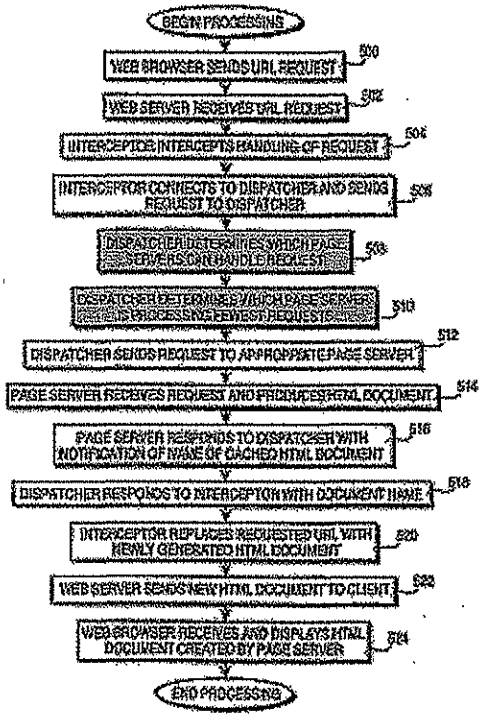
55. Below on the left is a flow chart describing how a request is processed in the prior art Web server disclosed by the '554 Patent and on the right is a flow chart from Popp describing how it processes a request:



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 5

THIS EXHIBIT HAS BEEN REDACTED IN ITS ENTIRETY

Exhibit 6

THIS EXHIBIT HAS BEEN REDACTED IN ITS ENTIRETY

Exhibit 7

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
1(a): A computer-implemented method for managing a dynamic Web page generation request to a Web server, said computer-implemented method comprising the steps of:	"The Oracle WebServer is an HTTP server with a tightly integrated Oracle7 server that enables the creation of dynamic HTML documents from data stored in an Oracle database." MAC003052	Dr. Shamos did not prove that Reference 153 (Ex. 44) meets the preamble. Furthermore, as addressed earlier in my report, this reference merely describes an HTTP server executing a CGI program to call a database backend program to generate an HTML page. Oracle readily acknowledged to the industry that "clearly the way to go is not to use CGI, so you can avoid that performance bottleneck." (Ex. 51, ORCL01606516; Ex. 53, Montinola Dep. at 160). Oracle recognized that CGI "has limitations, especially in performance and scalability." (Ex. 49, ORCL937793; Ex. 53, Montinola Dep. at 178-179). These are the same shortcomings of CGI disclosed in the specification of the '554 patent.
1(b)(1): routing said request from said Web server to a page server	"The Oracle7 Server provides the storage for all dynamic data in relational tables, and all the program logic used to create dynamic HTML pages." MAC003052. Thus the Oracle7 Server is a "page server" of the patent. The dynamic page generation request is received by the web listener (web server) and routed to the Oracle7 server, as shown in the diagram on page MAC003053.	Under Oracle's proposed construction, the "page server" is "page generating software, running on a processor separate from that of the Web server, that generates dynamic Web pages." Dr. Shamos failed to prove that Reference 153 discloses a page server under Oracle's proposed construction because he failed to identify a "page server" that executes on a processor separate from the processor of the Web Server.
1(b)(2): said page server receiving said request and	The diagram on page MAC003053 shows that the request is received by the Oracle7 Server (page server).	
1(b)(3): releasing said Web server to process other requests	"For maximum performance the Web Listener is designed to run as an asynchronous engine … providing high performance under a heavy load." p. 3-2. Here "asynchronous" means that it functions	Dr. Shamos contended that "TCP Releasing" and "SBL Releasing" disclosed by Reference 153 meets this limitation under either party's claim construction. I agree.

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
	independently of other tasks, which are being performed simultaneously. For the web listener (web server) to run asynchronously, it must be released to respond to other requests after passing off a request to a page server. Further, OIS passes requests using HTTP, so TCP releasing is performed. Page 1-3 shows SQL*Net running over a TCP/IP connection. The "releasing limitation" is met under the TCP theory of releasing. Furthermore, the Oracle7 server binds to a TCP port and therefore the releasing limitation is met under the SBL theory of releasing. MAC003053	
1(b)(4): wherein said routing step further includes the steps of intercepting said request at said Web server	"When the Oracle Web Listener receives a request from a client, it first determines whether that request is for a static document or a dynamic document. If the request is for a static document, the Web Listener sends the file and the associated type information directly to the client. If the request is for a dynamic document, it is created 'on the fly' by a program invoked by the Web Listener, in compliance with the Common Gateway Interface (CGI)." MAC003053. Thus the Web Listener performs interception	
1(b)(5): routing said request from said Web server to a dispatcher, and	The web agent in OracleServer acts as a dispatcher. "The Oracle Web Agent is a CGI program that the Web Listener executes when a request is received for a dynamic document. To connect to different servers, or different schemas on the same server, multiple Web Agent services may be configured on a single Oracle WebServer." MAC003054. Thus the Web Agent makes an informed choice of which Web Agent service is to be invoked to handle the request.	As discussed in the section below, Dr. Shamos failed to establish that Reference 153 has a "dispatcher" because, as shown below, Reference 153 does not disclose "dispatching" under either Oracle's or epicRealm's proposed construction. As explained below, the Web Agent cannot be a dispatcher because the Web Agent has no choice of which Oracle7 Server to connect to. (Ex. 53, Montinola Dep. at 81). The Web agent merely follows the instructions contained in

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
	"When a request from a Web browser comes in, the Web Listener will extract the service name that is embedded in the URL and find out which parameters to use by reading the owa.cfg file." MAC003148. This is configuration file dispatching.	the URL in conjunction with a configuration file. (Ex. 53, Montinola Dep. at 61-62, 80-81). Furthermore, Dr. Shamos failed to identify any dynamic information that the Web Agent maintains about page servers indicating which page server can more efficiently process the request.
1(b)(6): dispatching said request to said page server	Oracle WebServer 1.0 also performs SQL*Net dispatching. Page 1-3. "To connect to the Oracle7 Server, the Web Agent needs the following information to be specified in the Web Agent service: • username • password • ORACLE_HOME • ORACLE_SID (for local databases only) • SQL*Net V2 Service Name or Connect String (for remote databases only). The Administration Utility allows the administrator to display, create, modify or delete a Web Agent service. With the Web Agent Creation form, you do not need to modify the configuration file for the Oracle Web Agent service (owa.cfg) directly." MAC003148-3150.	Reference 153 does not disclose "dispatching" under either Oracle's or epicRealm's proposed construction. First, Dr. Shamos contended that his "dispatcher" of Reference 153 (Web Agent) makes an informed selection of which "page server" to use. However, Dr. Shamos did not distinguish between epicRealm's construction and Oracle's construction. EpicRealm's proposed construction for "dispatching" is "examining a request to make an informed selection of which page server should process the request based on dynamic information maintained about page servers, the dynamic information indicating which page server can more efficiently process the request, and sending the request to the selected page server." Dr. Shamos failed to identify any dynamic information maintained about page servers indicating which page server can more efficiently process the request. The configuration file does not maintain dynamic information because the configuration file contains static information and is manually changed by a programmer, not automatically changed. I address Dr. Shamos's contention regarding "Configuration File Dispatching" earlier in my report. Furthermore, the configuration file does not maintain information about which page server can more efficiently process a request. The configuration file

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
		exists in storage memory of the server and specifies database connection information for the *only* database to process the request. The configuration file does not distinguish between the relative efficiencies of page servers. Thus, Reference 153 does not disclose "dispatching" under epicRealm's proposed construction. Similarly, Dr. Shamos failed to demonstrate that Reference 153 discloses "dispatching" under Oracle's construction. At a minimum, Dr. Shamos did not distinguish his analysis under epicRealm's and Oracle's proposed construction for "dispatching." Therefore, for the reasons set forth above, Reference 153 failed to disclose "dispatching." In addition, Oracle's construction requires that the dispatcher make an "informed selection of which page server should process the request...." In Reference 153, there is not an "informed selection" because the request is routed to the database that is specified in the configuration file based on the URL. In other words, the dispatcher is not making a selection from the available "page servers." Oracle's corporate witness on Oracle WebServer 1.0 testified that the Web Agent has no choice of which Oracle7 Server to connect to. (Ex. 53, Montinola Dep. at 81). The Web agent merely follows the instructions contained in the URL in conjunction with a configuration file. (Ex. 53, Montinola Dep. at 61-62, 80-81). The Web Agent also has no real-time status information about the Oracle7 Server. (Ex. 53, Montinola Dep. at 81). Oracle, who considered itself a market leader who released state-of-the-art products, readily acknowledged that prior to its release of Oracle

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
		Webserver 2.0, Oracle was not aware of any web servers that used load balancing or any programs that generated dynamic web pages that would have used any type of load balancing. (Ex. 53, Montinola Dep. at 221). Thus, Reference 153 does not disclose "dispatching" under Oracle's proposed construction.
1(c): processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests, and	The Oracle7 server (page server) processes the dynamic page generation request and returns the generated page to the web agent. This process occurs simultaneously with the web listener (web server) processing other requests because the Web Listener operates asynchronously. "The Oracle Web Listener can handle a large number of simultaneous requests, and has advanced features which use system resources more efficiently than other HTTP servers available on the market." MAC003053. Also, OWS 1.0 allows for instantiation of multiple web listeners, allowing for concurrent processing of requests.	
1(d)(1): dynamically generating a Web page in response to said request	See '554 1c	
1(d)(2): said Web page including data dynamically retrieved from one or more data sources.	The generated web page includes data dynamically retrieved from one or more data sources by means of the PL/SQL procedures, which access an SQL database. "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." MAC003054.	

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
2: The computer-implemented method in claim 1 wherein said step of processing said request includes the step of identifying said one or more data sources from which to retrieve said data.	"After connecting to the database the Web Agent invokes the appropriate PL/SQL procedure, whose name is obtained by parsing the PATH_INFO environment variable." MAC003146. The data sources to be used are identified by examining the PATH_INFO environment variable.	Reference 153 cannot anticipate this claim because it does not anticipate the claim from which this claim depends. Dr. Shamos failed to demonstrate how Reference 153 meets this limitation. Dr. Shamos merely identified that the "web agent" identifies the database through PATH_INFO, however, Dr. Shamos contended that "web agent" is the dispatcher. The limitation requires that the page server that performs the processing also identifies the data sources. Therfore, Dr. Shamos failed to establish that Reference 153 meets this limitation. Moreover, Oracle's construction requires "listing in an HTML extension tag the location of said one or more data sources." Dr. Shamos did not contend that Reference 153 discloses such an HTML extension tag.
3: The computer-implemented method in claim 2 wherein said step of dynamically generating said Web page includes the step of dynamically retrieving said data from said one or more data sources.	"The Oracle WebServer is an HTTP server with a tightly integrated Oracle7 Server that enables the creation of dynamic HTML documents from data stored in an Oracle database. When the data changes, these HTML documents are updated automatically….This approach supplements the presentation of static, or unchanging, data which is found on most sites today, with the dynamic real-time data present in business systems based on the Oracle 7 Server." p. 1-2 (MAC003052). 5. The PL/SQL procedure executes, generating an HTML document. With the help of the Developer's Toolkit, the PL/SQL procedure extracts data from the Oracle7	Reference 153 cannot anticipate this claim because it does not anticipate the claim from which this claim depends.

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
	database and generates an HTML document in a PL/SQL table." (Oracle, p. 5-5).	
4: The computer-implemented method in claim 3 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more data sources.	Connection caching is used in SQL*Net, which may optionally be used with OWS 1.0. ORCL01806717	Reference 153 cannot anticipate this claim because it does not anticipate the claim from which this claim depends. The citation failed to support Dr. Shamos's contention. First, I could find no reference to connection caching or Oracle 1.0 on ORCL01806717. (Ex. 50.) Second, Dr. Shamos failed to identify any evidence supporting his statement that connection caching in SQL*NET may optionally be used with Oracle 1.0. Therefore, Dr. Shamos failed to show that Reference 153 meets this limitation.
5: The computer-implemented method in claim 3 wherein said step of processing said request includes the step of logging into said one or more data sources.	"In order to connect to an Oracle 7 Server, the Web Agent requires certain information, such as which server to connect to and what username and password to use." (MAC003146). The username and password are used to log in to the data source. "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." (p. 1-4 (MAC 3054; ORCL 23). "When the Oracle Web Agent logs into the Oracle7 Server, it starts a PL/SQL procedure that has	Reference 153 cannot anticipate this claim because it does not anticipate the claim from which this claim depends. Dr. Shamos failed to demonstrate how Reference 153 meets this limitation. Dr. Shamos merely identified that the "web agent" logs into a data source, however, Dr. Shamos contended that "web agent" is the dispatcher. The limitation requires that the page server that performs the logging into the data sources. Therefore, Dr. Shamos failed to establish that Reference 153 meets this limitation.

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
	been created by the user to generate an HTML page as its output." (p. 1-4 (MAC 3054; ORCL 23).)	
6: The computer-implemented method in claim 3 wherein said step of dynamically generating said Web page includes the step of maintaining a page cache containing said Web page.	Pages MAC003097-98, MAC003223-24 and MAC003255-56 explain how to set up file caching, which "decreases the access time required for commonly-used files." "The Oracle Web Listener allows a configurable set of commonly accessed files to be cached in memory. This provides very good performance when these files are accessed by a client." MAC003086	Not asserted.
7: The computer-implemented method in claim 3 wherein said page server includes custom HTML extension templates for configuring said Web page.	An extension template is a construct that is not HTML but that can easily be converted into HTML. It "extends" HTML by providing the user with additional facilities. Oracle Webserver provides a Developer's Toolkit incorporating hypertext procedures (HTP), described in Section 6: "A hypertext procedure generates a line in an HTML document that contains the HTML tag that corresponds to its name. For instance, the htp.anchor procedure generates an anchor tag." A variety of hypertext procedures are provided with the toolkit, including "Body Tags," "List Tags," "Form Tags" and "Table Tags" which "allow the user to insert tables and manipulate the size and columns of the table in a document. All of these are HTML extension templates	Reference 153 cannot anticipate this claim because it does not anticipate the claim from which this claim depends. Dr. Shamos failed to demonstrate how Reference 153 meets this limitation. Reference 153 does not disclose custom HTML templates as they are understood by one of ordinary skill in the art. What Dr. Shamos contended are "custom HTML templates" are just procedures that are used to generate HTML tags. Even under Dr. Shamos's definition, these procedures are not "custom HTML templates."
8: The computer-implemented method in claim 7 wherein said step of processing said request further includes the step of	The reference gives many examples of inserting dynamic data obtained by accessing an SQL database into an HTML extension template. For example, the procedure htp.cite, described on page 6-20 of the	Reference 153 cannot anticipate this claim because it does not anticipate the claim from which this claim depends.

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
inserting said dynamically retrieved data from said one or more data sources into said custom HTML extension templates.	reference (MAC003182), when called with two arguments, e.g. htp.cite(ctext, cattributes), causes the arguments to be substituted into a template as follows: <CITE cattributes>ctext</CITE>. See PL/SQL procedures at MAC003155-3157	Dr. Shamos failed to demonstrate how Reference 153 meets this limitation. As explained above, Dr. Shamos failed to explain how Reference 153 discloses "customer HTML templates." In addition, Dr. Shamos's contention is inconsistent with his contention in claim 7. Even if Dr. Shamos's contention is correct regarding the hypertext procedure being a "customer HTML template" in claim 7, which it is not, the htp.cite function relied on by Dr. Shamos in claim 8 is not a hyper text procedure which Dr. Shamos identified in claim 7.
9(a): A networked system for managing a dynamic Web page generation request, said system comprising	OIS operates with a Web server, hence is networked	Dr. Shamos did not prove that Reference 153 meets the preamble. Furthermore, as addressed earlier in my report, this reference merely describes an HTTP server executing a CGI program to call a database backend program to generate an HTML page. Oracle readily acknowledged to the industry that "clearly the way to go is not to use CGI, so you can avoid that performance bottleneck." (Ex. 51, ORCL01606516; Ex. 53, Montinola Dep. at 160). Oracle recognized that CGI "has limitations, especially in performance and scalability." (Ex. 49, ORCL937793; Ex. 53, Montinola Dep. at 178-179). These are the same shortcomings of CGI disclosed in the specification of the '554 patent.
9(b): one or more data sources	The generated web page includes data dynamically retrieved from one or more data sources by means of the PL/SQL procedures, which access an SQL database. "To connect to different servers, or different schemas on the same server, multiple Web Agent services may be	

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
	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." MAC003054.	
9(c): a page server having a processing means	"The Oracle7 Server provides the storage for all dynamic data in relational tables, and all the program logic used to create dynamic HTML pages." MAC003052. Thus the Oracle7 Server is a "page server" of the patent. The dynamic page generation request is received by the web listener (web server) and routed to the Oracle7 server, as shown in the diagram on page MAC003053. "The Oracle7 Server provides the storage for all dynamic data in relational tables, and all the program logic used to create dynamic HTML pages." MAC003052. Thus the Oracle7 Server is a "page server" of the patent. A processor of a computer that runs Oracle7 server is a page server processing means.	Under Oracle's proposed construction, the "page server" is "page generating software, running on a processor separate from that of the second computer system, that generates dynamic Web pages." Dr. Shamos failed to prove that Reference 153 discloses a page server under Oracle's proposed construction because he failed to identify a "page server" that executes on a processor separate from the processor of the second computer system. Dr. Shamos failed to identify the specified function and the corresponding structure, material or acts for the "processing means" as required by 35 U.S.C. § 112 ¶ 6. Furthermore, Dr. Shamos failed to demonstrate that Reference 153 that discloses the specified function and corresponding structure, material or acts for the "processing means" or equivalents thereof.
9(d): a first computer system including means for generating said request, and	The first computer system is a computer that has a processor that runs a web browser as shown on page 1-3 (MAC003053).	Dr. Shamos failed to identify the specified function and the corresponding structure, material or acts for the "means for generating said request" as required by 35 U.S.C. § 112 ¶ 6. Furthermore, Dr. Shamos failed to demonstrate that Reference 153 that discloses the specified function and corresponding structure, material or acts for the "means for generating said request" or equivalents thereof.
9(e)(1): a second computer system including	The second computer system is a computer having a processor that runs the Web Listener is shown in a	

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
	rectangular box on p. 1-3, MAC003053.	
9(e)(2): means for receiving said request from said first computer	"The Oracle7 Server provides the storage for all dynamic data in relational tables, and all the program logic used to create dynamic HTML pages." MAC003052. Thus the Oracle7 Server is a "page server" of the patent. The dynamic page generation request is received by the web listener (web server) and routed to the Oracle7 server, as shown in the diagram on page MAC003053. The means is a processor of a computer that runs the Web Listener. See figure on MAC003053.	Dr. Shamos failed to identify the specified function and the corresponding structure, material or acts for the "means for receiving" as required by 35 U.S.C. § 112 ¶ 6. Furthermore, Dr. Shamos failed to demonstrate that Reference 194 that discloses the specified function and corresponding structure, material or acts for the "means for receiving" or equivalents thereof.
9(e)(3): said second computer system also including a router, said router routing said request from said second computer system to said page server	"The Oracle7 Server provides the storage for all dynamic data in relational tables, and all the program logic used to create dynamic HTML pages." MAC003052. Thus the Oracle7 Server is a "page server" of the patent. The dynamic page generation request is received by the web listener (web server) and routed to the Oracle7 server, as shown in the diagram on page MAC003053.	Dr. Shamos failed to identify a "router."
9(e)(4): wherein said routing further includes intercepting said request at said second computer	"When the Oracle Web Listener receives a request from a client, it first determines whether that request is for a static document or a dynamic document. If the request is for a static document, the Web Listener sends the file and the associated type information directly to the client. If the request is for a dynamic document, it is created 'on the fly' by a program invoked by the Web Listener, in compliance with the Common Gateway Interface (CGI)." MAC003053. Thus the Web Listener performs interception	

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
9(e)(5): routing said request from said second computer to a dispatcher, and	The web agent in OracleServer acts as a dispatcher. "The Oracle Web Agent is a CGI program that the Web Listener executes when a request is received for a dynamic document. To connect to different servers, or different schemas on the same server, multiple Web Agent services may be configured on a single Oracle WebServer." MAC003054. Thus the Web Agent makes an informed choice of which Web Agent service is to be invoked to handle the request. "When a request from a Web browser comes in, the Web Listener will extract the service name that is embedded in the URL and find out which parameters to use by reading the owa.cfg file." MAC003148. This is configuration file dispatching.	As discussed in the section below, Dr. Shamos failed to establish that Reference 153 has a "dispatcher" because, as shown below, Reference 153 does not disclose "dispatching" under either Oracle's or epicRealm's proposed construction. As explained below, the Web Agent cannot be a dispatcher because the Web Agent has no choice of which Oracle7 Server to connect to. (Ex. 53, Montinola Dep. at 81). The Web agent merely follows the instructions contained in the URL in conjunction with a configuration file. (Ex. 53, Montinola Dep. at 61-62, 80-81). Furthermore, Dr. Shamos failed to identify any dynamic information that the Web Agent maintains about page servers indicating which page server can more efficiently process the request.
9(e)(6): dispatching said request to said page server	Oracle WebServer 1.0 also performs SQL*Net dispatching. Page 1-3. "To connect to the Oracle7 Server, the Web Agent needs the following information to be specified in the Web Agent service: • username • password • ORACLE_HOME • ORACLE_SID (for local databases only) • SQL*Net V2 Service Name or Connect String (for remote databases only). The Administration Utility allows the administrator to display, create, modify or delete a Web Agent service. With the Web Agent Creation form, you do not need to modify the configuration file for the Oracle Web Agent service (owa.cfg) directly." MAC003148-3150.	Reference 153 does not disclose "dispatching" under either Oracle's or epicRealm's proposed construction. First, Dr. Shamos contended that his "dispatcher" of Reference 153 (Web Agent) makes an informed selection of which "page server" to use. However, Dr. Shamos did not distinguish between epicRealm's construction and Oracle's construction. EpicRealm's proposed construction for "dispatching" is "examining a request to make an informed selection of which page server should process the request based on dynamic information maintained about page servers, the dynamic information indicating which page server can more efficiently process the request, and sending the request to the selected page server." Dr. Shamos failed to identify any dynamic information maintained about page servers indicating which page server can more

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
		efficiently process the request. The configuration file does not maintain dynamic information because the configuration file contains static information and is manually changed by a programmer, not automatically changed. I address Dr. Shamos's contention regarding "Configuration File Dispatching" earlier in my report. Furthermore, the configuration file does not maintain information about which page server can more efficiently process a request. The configuration file exists in storage memory of the server and specifies database connection information for the *only* database to process the request. The configuration file does not distinguish between the relative efficiencies of page servers. Thus, Reference 153 does not disclose "dispatching" under epicRealm's proposed construction. Similarly, Dr. Shamos failed to demonstrate that Reference 153 discloses "dispatching" under Oracle's construction. At a minimum, Dr. Shamos did not distinguish his analysis under epicRealm's and Oracle's proposed construction for "dispatching." Therefore, for the reasons set forth above, Reference 153 failed to disclose "dispatching." In addition, Oracle's construction requires that the dispatcher make an "informed selection of which page server should process the request...." In Reference 153, there is not an "informed selection" because the request is routed to the database that is specified in the configuration file based on the URL. In other words, the dispatcher is not making a selection from the available "page servers." Oracle's corporate witness on Oracle WebServer 1.0 testified that the Web Agent has no

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
		choice of which Oracle7 Server to connect to. (Ex. 53, Montinola Dep. at 81). The Web agent merely follows the instructions contained in the URL in conjunction with a configuration file. (Ex. 53, Montinola Dep. at 61-62, 80-81). The Web Agent also has no real-time status information about the Oracle7 Server. (Ex. 53, Montinola Dep. at 81). Oracle, who considered itself a market leader who released state-of-the-art products, readily acknowledged that prior to its release of Oracle Webserver 2.0, Oracle was not aware of any web servers that used load balancing or any programs that generated dynamic web pages that would have used any type of load balancing. (Ex. 53, Montinola Dep. at 221). Thus, Reference 153 does not disclose "dispatching" under Oracle's proposed construction.
9(e)(7): said page server receiving said request and	The diagram on page MAC003053 shows that the request is received by the Oracle7 Server (page server).	
9(e)(8): releasing said second computer system to process other requests	"For maximum performance the Web Listener is designed to run as an asynchronous engine … providing high performance under a heavy load." p. 3-2. Here "asynchronous" means that it functions independently of other tasks, which are being performed simultaneously. For the web listener (web server) to run asynchronously, it must be released to respond to other requests after passing off a request to a page server. Further, OIS passes requests using HTTP, so TCP releasing is performed. Page 1-3 shows SQL*Net running over a TCP/IP connection. The "releasing limitation" is met under the TCP theory of releasing. Furthermore, the Oracle7 server	Dr. Shamos contended that "TCP Releasing" and "SBL Releasing" disclosed by Reference 153 meets this limitation under either party's claim construction. I agree.

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
	binds to a TCP port and therefore the releasing limitation is met under the SBL theory of releasing. MAC003053	
9(e)(9): said page server processing means processing said request and dynamically generating a Web page in response to said request	See '554 1c	
9(e)(10): said Web page including data dynamically retrieved from said one or more data sources	The generated web page includes data dynamically retrieved from one or more data sources by means of the PL/SQL procedures, which access an SQL database. "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." MAC003054.	
10(a): The networked system in claim 9 wherein said router in said second computer system includes:	OIS operates with a Web server, hence is networked	Reference 153 cannot anticipate this claim because it does not anticipate the claim from which this claim depends. Dr. Shamos does not contend nor demonstrate that Reference 153 meets the preamble of claim 10 of the '554 patent.
10(b): an interceptor intercepting said request at said second computer system and routing said request, and	"When the Oracle Web Listener receives a request from a client, it first determines whether that request is for a static document or a dynamic document. If the request is for a static document, the Web Listener sends the file and the associated type information	Reference 153 cannot anticipate this claim because it does not anticipate the claim from which this claim depends.

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
	directly to the client. If the request is for a dynamic document, it is created 'on the fly' by a program invoked by the Web Listener, in compliance with the Common Gateway Interface (CGI)." MAC003053. Thus the Web Listener performs interception	
10(c): a dispatcher receiving said routed request from said interceptor and dispatching said request to said page server	Oracle WebServer 1.0 also performs SQL*Net dispatching. Page 1-3. "To connect to the Oracle7 Server, the Web Agent needs the following information to be specified in the Web Agent service: • username • password • ORACLE_HOME • ORACLE_SID (for local databases only) • SQL*Net V2 Service Name or Connect String (for remote databases only). The Administration Utility allows the administrator to display, create, modify or delete a Web Agent service. With the Web Agent Creation form, you do not need to modify the configuration file for the Oracle Web Agent service (owa.cfg) directly." MAC003148-3150.	Reference 153 cannot anticipate this claim because it does not anticipate the claim from which this claim depends. Reference 153 does not disclose "dispatching" under either Oracle's or epicRealm's proposed construction. First, Dr. Shamos contended that his "dispatcher" of Reference 153 (Web Agent) makes an informed selection of which "page server" to use. However, Dr. Shamos did not distinguish between epicRealm's construction and Oracle's construction. EpicRealm's proposed construction for "dispatching" is "examining a request to make an informed selection of which page server should process the request based on dynamic information maintained about page servers, the dynamic information indicating which page server can more efficiently process the request, and sending the request to the selected page server." Dr. Shamos failed to identify any dynamic information maintained about page servers indicating which page server can more efficiently process the request. The configuration file does not maintain dynamic information because the configuration file contains static information and is manually changed by a programmer, not automatically changed. I address Dr. Shamos's contention regarding "Configuration File Dispatching" earlier in my report.

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
		Furthermore, the configuration file does not maintain information about which page server can more efficiently process a request. The configuration file exists in storage memory of the server and specifies database connection information for the *only* database to process the request. The configuration file does not distinguish between the relative efficiencies of page servers. Thus, Reference 153 does not disclose "dispatching" under epicRealm's proposed construction. Similarly, Dr. Shamos failed to demonstrate that Reference 153 discloses "dispatching" under Oracle's construction. At a minimum, Dr. Shamos did not distinguish his analysis under epicRealm's and Oracle's proposed construction for "dispatching." Therefore, for the reasons set forth above, Reference 153 failed to disclose "dispatching." In addition, Oracle's construction requires that the dispatcher make an "informed selection of which page server should process the request...." In Reference 153, there is not an "informed selection" because the request is routed to the database that is specified in the configuration file based on the URL. In other words, the dispatcher is not making a selection from the available "page servers." Oracle's corporate witness on Oracle WebServer 1.0 testified that the Web Agent has no choice of which Oracle7 Server to connect to. (Ex. 53, Montinola Dep. at 81). The Web agent merely follows the instructions contained in the URL in conjunction with a configuration file. (Ex. 53, Montinola Dep. at 61-62, 80-81). The Web Agent also has no real-time status information about the Oracle7 Server. (Ex. 53,

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
		Montinola Dep. at 81). Oracle, who considered itself a market leader who released state-of-the-art products, readily acknowledged that prior to its release of Oracle Webserver 2.0, Oracle was not aware of any web servers that used load balancing or any programs that generated dynamic web pages that would have used any type of load balancing. (Ex. 53, Montinola Dep. at 221). Thus, Reference 153 does not disclose "dispatching" under Oracle's proposed construction.
11(a): A machine readable medium having stored thereon data representing sequences of instructions, which when executed by a computer system, cause said computer system to perform the steps of:	OIS is a programmed system whose instructions are on a machine readable medium	Dr. Shamos did not prove that Reference 153 meets the preamble.
11(b)(1): routing a dynamic Web page generation request from a Web server to a page server	"The Oracle7 Server provides the storage for all dynamic data in relational tables, and all the program logic used to create dynamic HTML pages." MAC003052. Thus the Oracle7 Server is a "page server" of the patent. The dynamic page generation request is received by the web listener (web server) and routed to the Oracle7 server, as shown in the diagram on page MAC003053.	Under Oracle's proposed construction, the "page server" is "page generating software, running on a processor separate from that of the Web server, that generates dynamic Web pages." Dr. Shamos failed to prove that Reference 153 discloses a page server under Oracle's proposed construction because he failed to identify a "page server" that executes on a processor separate from the processor of the Web Server.
11(b)(2): said page server receiving said request and	The diagram on page MAC003053 shows that the request is received by the Oracle7 Server (page server).	
11(b)(3): releasing said Web server to process other requests	"For maximum performance the Web Listener is designed to run as an asynchronous engine … providing high performance under a heavy load." p.	Dr. Shamos contended that "TCP Releasing" and "SBL Releasing" disclosed by Reference 153 meets this limitation under either party's claim construction. I

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
	3-2. Here "asynchronous" means that it functions independently of other tasks, which are being performed simultaneously. For the web listener (web server) to run asynchronously, it must be released to respond to other requests after passing off a request to a page server. Further, OIS passes requests using HTTP, so TCP releasing is performed. Page 1-3 shows SQL*Net running over a TCP/IP connection. The "releasing limitation" is met under the TCP theory of releasing. Furthermore, the Oracle7 server binds to a TCP port and therefore the releasing limitation is met under the SBL theory of releasing. MAC003053	agree.
11(b)(4): wherein said routing step further includes the steps of intercepting said request at said Web server	"When the Oracle Web Listener receives a request from a client, it first determines whether that request is for a static document or a dynamic document. If the request is for a static document, the Web Listener sends the file and the associated type information directly to the client. If the request is for a dynamic document, it is created 'on the fly' by a program invoked by the Web Listener, in compliance with the Common Gateway Interface (CGI)." MAC003053. Thus the Web Listener performs interception	
11(b)(5): routing said request from said Web server to a dispatcher, and	The web agent in OracleServer acts as a dispatcher. "The Oracle Web Agent is a CGI program that the Web Listener executes when a request is received for a dynamic document. To connect to different servers, or different schemas on the same server, multiple Web Agent services may be configured on a single Oracle WebServer." MAC003054. Thus the Web Agent makes an informed choice of which Web	As discussed in the section below, Dr. Shamos failed to establish that Reference 153 has a "dispatcher" because, as shown below, Reference 153 does not disclose "dispatching" under either Oracle's or epicRealm's proposed construction. As explained below, the Web Agent cannot be a dispatcher because the Web Agent has no choice of which Oracle7 Server to connect to. (Ex. 53, Montinola Dep. at 81). The

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
	Agent service is to be invoked to handle the request. "When a request from a Web browser comes in, the Web Listener will extract the service name that is embedded in the URL and find out which parameters to use by reading the owa.cfg file." MAC003148. This is configuration file dispatching.	Web agent merely follows the instructions contained in the URL in conjunction with a configuration file. (Ex. 53, Montinola Dep. at 61-62, 80-81). Furthermore, Dr. Shamos failed to identify any dynamic information that the Web Agent maintains about page servers indicating which page server can more efficiently process the request.
11(b)(6): dispatching said request to said page server	Oracle WebServer 1.0 also performs SQL*Net dispatching. Page 1-3. "To connect to the Oracle7 Server, the Web Agent needs the following information to be specified in the Web Agent service: • username • password • ORACLE_HOME • ORACLE_SID (for local databases only) • SQL*Net V2 Service Name or Connect String (for remote databases only). The Administration Utility allows the administrator to display, create, modify or delete a Web Agent service. With the Web Agent Creation form, you do not need to modify the configuration file for the Oracle Web Agent service (owa.cfg) directly." MAC003148-3150.	Reference 153 does not disclose "dispatching" under either Oracle's or epicRealm's proposed construction. First, Dr. Shamos contended that his "dispatcher" of Reference 153 (Web Agent) makes an informed selection of which "page server" to use. However, Dr. Shamos did not distinguish between epicRealm's construction and Oracle's construction. EpicRealm's proposed construction for "dispatching" is "examining a request to make an informed selection of which page server should process the request based on dynamic information maintained about page servers, the dynamic information indicating which page server can more efficiently process the request, and sending the request to the selected page server." Dr. Shamos failed to identify any dynamic information maintained about page servers indicating which page server can more efficiently process the request. The configuration file does not maintain dynamic information because the configuration file contains static information and is manually changed by a programmer, not automatically changed. I address Dr. Shamos's contention regarding "Configuration File Dispatching" earlier in my report. Furthermore, the configuration file does not maintain information about which page server can more

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
		efficiently process a request. The configuration file exists in storage memory of the server and specifies database connection information for the *only* database to process the request. The configuration file does not distinguish between the relative efficiencies of page servers. Thus, Reference 153 does not disclose "dispatching" under epicRealm's proposed construction. Similarly, Dr. Shamos failed to demonstrate that Reference 153 discloses "dispatching" under Oracle's construction. At a minimum, Dr. Shamos did not distinguish his analysis under epicRealm's and Oracle's proposed construction for "dispatching." Therefore, for the reasons set forth above, Reference 153 failed to disclose "dispatching." In addition, Oracle's construction requires that the dispatcher make an "informed selection of which page server should process the request...." In Reference 153, there is not an "informed selection" because the request is routed to the database that is specified in the configuration file based on the URL. In other words, the dispatcher is not making a selection from the available "page servers." Oracle's corporate witness on Oracle WebServer 1.0 testified that the Web Agent has no choice of which Oracle7 Server to connect to. (Ex. 53, Montinola Dep. at 81). The Web agent merely follows the instructions contained in the URL in conjunction with a configuration file. (Ex. 53, Montinola Dep. at 61-62, 80-81). The Web Agent also has no real-time status information about the Oracle7 Server. (Ex. 53, Montinola Dep. at 81). Oracle, who considered itself a market leader who released state-of-the-art products,

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
		readily acknowledged that prior to its release of Oracle Webserver 2.0, Oracle was not aware of any web servers that used load balancing or any programs that generated dynamic web pages that would have used any type of load balancing. (Ex. 53, Montinola Dep. at 221). Thus, Reference 153 does not disclose "dispatching" under Oracle's proposed construction.
11(c): processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests, and	The Oracle7 server (page server) processes the dynamic page generation request and returns the generated page to the web agent. This process occurs simultaneously with the web listener (web server) processing other requests because the Web Listener operates asynchronously. "The Oracle Web Listener can handle a large number of simultaneous requests, and has advanced features which use system resources more efficiently than other HTTP servers available on the market." MAC003053. Also, OWS 1.0 allows for instantiation of multiple web listeners, allowing for concurrent processing of requests.	
11(d): dynamically generating a Web page, said Web page including data retrieved from one or more data sources	See '554 1c	

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
2[1(a)]: A computer-implemented method for managing a dynamic Web page generation request to a	"The Oracle WebServer is an HTTP server with a tightly integrated Oracle7 server that enables the creation of dynamic HTML documents from data stored in an Oracle database." MAC003052	Dr. Shamos did not prove that Reference 153 meets the preamble. Furthermore, as addressed earlier in my report, this

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
Web server, said computer-implemented method comprising the steps of:		reference merely describes an HTTP server executing a CGI program to call a database backend program to generate an HTML page. Oracle readily acknowledged to the industry that "clearly the way to go is not to use CGI, so you can avoid that performance bottleneck." (Ex. 51, ORCL01606516; Ex. 53, Montinola Dep. at 160). Oracle recognized that CGI "has limitations, especially in performance and scalability." (Ex. 49, ORCL937793; Ex. 53, Montinola Dep. at 178-179). These are the same shortcomings of CGI disclosed in the specification of the '554 patent.
2[1(b)(1)]: routing a request from a Web server to a page server	"The Oracle7 Server provides the storage for all dynamic data in relational tables, and all the program logic used to create dynamic HTML pages." MAC003052. Thus the Oracle7 Server is a "page server" of the patent. The dynamic page generation request is received by the web listener (web server) and routed to the Oracle7 server, as shown in the diagram on page MAC003053.	Under Oracle's proposed construction, the "page server" is "page generating software, running on a processor separate from that of the Web server, that generates dynamic Web pages." Dr. Shamos failed to prove that Reference 153 discloses a page server under Oracle's proposed construction because he failed to identify a "page server" that executes on a processor separate from the processor of the Web Server.
2[1(b)(2)]: said page server receiving said request and	The diagram on page MAC003053 shows that the request is received by the Oracle7 Server (page server).	
2[1(b)(3)]: releasing said Web server to process other requests	"For maximum performance the Web Listener is designed to run as an asynchronous engine … providing high performance under a heavy load." p. 3-2. Here "asynchronous" means that it functions independently of other tasks, which are being performed simultaneously. For the web listener (web server) to run asynchronously, it must be released to respond to other requests after passing off a request	Dr. Shamos contended that "TCP Releasing" and "SBL Releasing" disclosed by Reference 153 meets this limitation under either party's claim construction. I agree.

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
	to a page server. Further, OIS passes requests using HTTP, so TCP releasing is performed. Page 1-3 shows SQL*Net running over a TCP/IP connection. The "releasing limitation" is met under the TCP theory of releasing. Furthermore, the Oracle7 server binds to a TCP port and therefore the releasing limitation is met under the SBL theory of releasing. MAC003053	
2[1(c)(1)]: wherein said routing step further includes the steps of: intercepting said request at said Web server and	"When the Oracle Web Listener receives a request from a client, it first determines whether that request is for a static document or a dynamic document. If the request is for a static document, the Web Listener sends the file and the associated type information directly to the client. If the request is for a dynamic document, it is created 'on the fly' by a program invoked by the Web Listener, in compliance with the Common Gateway Interface (CGI)." MAC003053. Thus the Web Listener performs interception	
2[1(c)(2)]: routing said request to said page server	"The Oracle7 Server provides the storage for all dynamic data in relational tables, and all the program logic used to create dynamic HTML pages." MAC003052. Thus the Oracle7 Server is a "page server" of the patent. The dynamic page generation request is received by the web listener (web server) and routed to the Oracle7 server, as shown in the diagram on page MAC003053.	Under Oracle's proposed construction, the "page server" is "page generating software, running on a processor separate from that of the Web server, that generates dynamic Web pages." Dr. Shamos failed to prove that Reference 153 discloses a page server under Oracle's proposed construction because he failed to identify a "page server" that executes on a processor separate from the processor of the Web Server.
2[1(d)]: processing said request, said processing being performed by said page server while said Web server	The Oracle7 server (page server) processes the dynamic page generation request and returns the generated page to the web agent. This process occurs simultaneously with the web listener (web server)	

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
concurrently processes said other requests, and	processing other requests because the Web Listener operates asynchronously. "The Oracle Web Listener can handle a large number of simultaneous requests, and has advanced features which use system resources more efficiently than other HTTP servers available on the market." MAC003053. Also, OWS 1.0 allows for instantiation of multiple web listeners, allowing for concurrent processing of requests.	
2[1(e)(1)]: dynamically generating a Web page in response to said request	See '554 1c	
2[1(e)(2)]: said Web page including data dynamically retrieved from one or more data sources	The generated web page includes data dynamically retrieved from one or more data sources by means of the PL/SQL procedures, which access an SQL database. "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." MAC003054.	
2(a): The computer-implemented method in claim 1 wherein said step of routing said request includes the steps of:	"The Oracle WebServer is an HTTP server with a tightly integrated Oracle7 server that enables the creation of dynamic HTML documents from data stored in an Oracle database." MAC003052	Dr. Shamos did not prove that Reference 153 meets the preamble.
2(b): routing said request from said Web server to a dispatcher, and	The web agent in OracleServer acts as a dispatcher. "The Oracle Web Agent is a CGI program that the Web Listener executes when a request is received for a dynamic document. To connect to different servers,	As discussed in the section below, Dr. Shamos failed to establish that Reference 153 has a "dispatcher" because, as shown below, Reference 153 does not disclose "dispatching" under either Oracle's or

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
	or different schemas on the same server, multiple Web Agent services may be configured on a single Oracle WebServer." MAC003054. Thus the Web Agent makes an informed choice of which Web Agent service is to be invoked to handle the request. "When a request from a Web browser comes in, the Web Listener will extract the service name that is embedded in the URL and find out which parameters to use by reading the owa.cfg file." MAC003148. This is configuration file dispatching.	epicRealm's proposed construction. As explained below, the Web Agent cannot be a dispatcher because the Web Agent has no choice of which Oracle7 Server to connect to. (Ex. 53, Montinola Dep. at 81). The Web agent merely follows the instructions contained in the URL in conjunction with a configuration file. (Ex. 53, Montinola Dep. at 61-62, 80-81). Furthermore, Dr. Shamos failed to identify any dynamic information that the Web Agent maintains about page servers indicating which page server can more efficiently process the request.
2(c): dispatching said request to said page server	Oracle WebServer 1.0 also performs SQL*Net dispatching. Page 1-3. "To connect to the Oracle7 Server, the Web Agent needs the following information to be specified in the Web Agent service: • username • password • ORACLE_HOME • ORACLE_SID (for local databases only) • SQL*Net V2 Service Name or Connect String (for remote databases only). The Administration Utility allows the administrator to display, create, modify or delete a Web Agent service. With the Web Agent Creation form, you do not need to modify the configuration file for the Oracle Web Agent service (owa.cfg) directly." MAC003148-3150.	Reference 153 does not disclose "dispatching" under either Oracle's or epicRealm's proposed construction. First, Dr. Shamos contended that his "dispatcher" of Reference 153 (Web Agent) makes an informed selection of which "page server" to use. However, Dr. Shamos did not distinguish between epicRealm's construction and Oracle's construction. EpicRealm's proposed construction for "dispatching" is "examining a request to make an informed selection of which page server should process the request based on dynamic information maintained about page servers, the dynamic information indicating which page server can more efficiently process the request, and sending the request to the selected page server." Dr. Shamos failed to identify any dynamic information maintained about page servers indicating which page server can more efficiently process the request. The configuration file does not maintain dynamic information because the configuration file contains static information and is manually changed by a programmer, not automatically

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
		changed. I address Dr. Shamos's contention regarding "Configuration File Dispatching" earlier in my report. Furthermore, the configuration file does not maintain information about which page server can more efficiently process a request. The configuration file exists in storage memory of the server and specifies database connection information for the *only* database to process the request. The configuration file does not distinguish between the relative efficiencies of page servers. Thus, Reference 153 does not disclose "dispatching" under epicRealm's proposed construction. Similarly, Dr. Shamos failed to demonstrate that Reference 153 discloses "dispatching" under Oracle's construction. At a minimum, Dr. Shamos did not distinguish his analysis under epicRealm's and Oracle's proposed construction for "dispatching." Therefore, for the reasons set forth above, Reference 153 failed to disclose "dispatching." In addition, Oracle's construction requires that the dispatcher make an "informed selection of which page server should process the request...." In Reference 153, there is not an "informed selection" because the request is routed to the database that is specified in the configuration file based on the URL. In other words, the dispatcher is not making a selection from the available "page servers." Oracle's corporate witness on Oracle WebServer 1.0 testified that the Web Agent has no choice of which Oracle7 Server to connect to. (Ex. 53, Montinola Dep. at 81). The Web agent merely follows the instructions contained in the URL in conjunction with a configuration file. (Ex. 53, Montinola Dep. at

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
		61-62, 80-81). The Web Agent also has no real-time status information about the Oracle7 Server. (Ex. 53, Montinola Dep. at 81). Oracle, who considered itself a market leader who released state-of-the-art products, readily acknowledged that prior to its release of Oracle Webserver 2.0, Oracle was not aware of any web servers that used load balancing or any programs that generated dynamic web pages that would have used any type of load balancing. (Ex. 53, Montinola Dep. at 221). Thus, Reference 153 does not disclose "dispatching" under Oracle's proposed construction.
3: The computer-implemented method in claim 1 wherein said step of processing said request includes the step of identifying said one or more data sources from which to retrieve said data	"After connecting to the database the Web Agent invokes the appropriate PL/SQL procedure, whose name is obtained by parsing the PATH_INFO environment variable." MAC003146. The data sources to be used are identified by examining the PATH_INFO environment variable.	Not asserted.
4: The computer-implemented method in claim 1 wherein said step of dynamically generating said Web page includes the step of dynamically retrieving said data from said one or more data sources	"The Oracle WebServer is an HTTP server with a tightly integrated Oracle7 Server that enables the creation of dynamic HTML documents from data stored in an Oracle database. When the data changes, these HTML documents are updated automatically….This approach supplements the presentation of static, or unchanging, data which is found on most sites today, with the dynamic real-time data present in business systems based on the Oracle 7 Server." p. 1-2 (MAC003052). 5. The PL/SQL procedure executes, generating an HTML document. With the help of the Developer's Toolkit, the PL/SQL procedure extracts data from the Oracle7 database and generates an HTML document in a	Not asserted.

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
	PL/SQL table." (Oracle, p. 5-5).	
5: The computer-implemented method in claim 1 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more data sources	Connection caching is used in SQL*Net, which may optionally be used with OWS 1.0. ORCL01806717	Not asserted.
6: The computer-implemented method in claim 1 wherein said step of processing said request includes the step of logging into said one or more data sources	"In order to connect to an Oracle 7 Server, the Web Agent requires certain information, such as which server to connect to and what username and password to use." (MAC003146). The username and password are used to log in to the data source. "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." (p. 1-4 (MAC 3054; ORCL 23). "When the Oracle Web Agent logs into the Oracle7 Server, it starts a PL/SQL procedure that has been created by the user to generate an HTML page as its output." (p. 1-4 (MAC 3054; ORCL 23).)	Not asserted.
7: The computer-implemented method in claim 1 wherein said step of dynamically generating said Web page includes the	Pages MAC003097-98, MAC003223-24 and MAC003255-56 explain how to set up file caching, which "decreases the access time required for commonly-used files." "The Oracle Web Listener	Not asserted.

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
step of maintaining a page cache containing said Web page	allows a configurable set of commonly accessed files to be cached in memory. This provides very good performance when these files are accessed by a client." MAC003086	
8: The computer-implemented method in claim 1 wherein said page server includes tag-based text templates for configuring said Web page	OIS uses HTML extension templates, which are tag-based. See '554 7	Not asserted.
9: The computer-implemented method in claim 8 wherein said step of processing said request further includes the step of inserting said-dynamically retrieved data from said one or more data sources into said tag-based text templates	See '554 8.	Not asserted.
10: The computer-implemented method in claim 8 wherein at least one of said tag-based text templates drives a format of the data dynamically retrieved from said one or more data sources in response to said request	Section 5 (e.g., page 5-5, MAC003147) describes the use of tag-based text templates in PL/SQL procedures to drive the format of the data in creating the generated page. "The Oracle WebServer Developer's Toolkit is a set of procedures that help the customer create programs to generate dynamic documents that are easily presented in HTML." p. 1-2 (MAC003052). HTML determines the format of a web page	Not asserted.
11: The computer-implemented method in claim 8 wherein said tag-based text templates include HTML templates	See '554 8.	Not asserted.

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
12: The computer-implemented method in claim 1 wherein said step of processing said request further includes the step of dynamically updating data at said one or more data sources	Dynamically updating the database is described on page 4-9 (MAC003113). "Which SQL statements determine the actual data content? Chiefly, three – INSERT, UPDATE, and DELETE. INSERT places rows in a table, UPDATE changes the values they contain, and DELETE removes them." p. 4-8 (MAC003112)	Not asserted.
13: The computer-implemented method in claim 1 wherein said step of processing said request further includes the step of processing an object handling extension	N/A	Not asserted.
14: The computer-implemented method in claim 13 wherein said object handling extension is an OLE extension	N/A	Not asserted.
16[15(a)]: A computer-implemented method comprising the steps of:	OIS is computer-implemented	Dr. Shamos did not prove that Reference 153 meets the preamble.
16[15(b)(1)]: transferring a request from an HTTP-compliant device to a page server	"The Oracle7 Server provides the storage for all dynamic data in relational tables, and all the program logic used to create dynamic HTML pages." MAC003052. Thus the Oracle7 Server is a "page server" of the patent. The dynamic page generation request is received by the web listener (web server) and routed to the Oracle7 server, as shown in the diagram on page MAC003053.	Under Oracle's proposed construction, the "page server" is "page generating software, running on a processor separate from that of the HTTP-compliant device, that generates dynamic Web pages." Dr. Shamos failed to prove that Reference 153 discloses a page server under Oracle's proposed construction because he failed to identify a "page server" that executes on a processor separate from the processor of the HTTP-compliant device.
16[15(b)(2)]: said page server receiving said request and	The diagram on page MAC003053 shows that the request is received by the Oracle7 Server (page	

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
	server).	
16[15(b)(3)]: releasing said HTTP-compliant device to process other requests	"For maximum performance the Web Listener is designed to run as an asynchronous engine … providing high performance under a heavy load." p. 3-2. Here "asynchronous" means that it functions independently of other tasks, which are being performed simultaneously. For the web listener (web server) to run asynchronously, it must be released to respond to other requests after passing off a request to a page server. Further, OIS passes requests using HTTP, so TCP releasing is performed. Page 1-3 shows SQL*Net running over a TCP/IP connection. The "releasing limitation" is met under the TCP theory of releasing. Furthermore, the Oracle7 server binds to a TCP port and therefore the releasing limitation is met under the SBL theory of releasing. MAC003053	Dr. Shamos contended that "TCP Releasing" and "SBL Releasing" disclosed by Reference 153 meets this limitation under either party's claim construction. I agree.
16[15(c)(1)]: wherein said transferring step further includes the steps of: intercepting said request at said HTTP-compliant device and	"When the Oracle Web Listener receives a request from a client, it first determines whether that request is for a static document or a dynamic document. If the request is for a static document, the Web Listener sends the file and the associated type information directly to the client. If the request is for a dynamic document, it is created 'on the fly' by a program invoked by the Web Listener, in compliance with the Common Gateway Interface (CGI)." MAC003053. Thus the Web Listener performs interception	
16[15(c)(2)]: transferring said request to said page server	"The Oracle7 Server provides the storage for all dynamic data in relational tables, and all the program logic used to create dynamic HTML pages."	Under Oracle's proposed construction, the "page server" is "page generating software, running on a processor separate from that of the HTTP-compliant

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
	MAC003052. Thus the Oracle7 Server is a "page server" of the patent. The dynamic page generation request is received by the web listener (web server) and routed to the Oracle7 server, as shown in the diagram on page MAC003053.	device, that generates dynamic Web pages." Dr. Shamos failed to prove that Reference 153 discloses a page server under Oracle's proposed construction because he failed to identify a "page server" that executes on a processor separate from the processor of the HTTP-complaint device.
16[15(d)]: processing said request, said processing being performed by said page server while said HTTP-compliant device concurrently processes said other requests, and	The Oracle7 server (page server) processes the dynamic page generation request and returns the generated page to the web agent. This process occurs simultaneously with the web listener (web server) processing other requests because the Web Listener operates asynchronously. "The Oracle Web Listener can handle a large number of simultaneous requests, and has advanced features which use system resources more efficiently than other HTTP servers available on the market." MAC003053. Also, OWS 1.0 allows for instantiation of multiple web listeners, allowing for concurrent processing of requests.	
16[15(e)(1)]: dynamically generating a page in response to said request	See '554 1c	
16[15(e)(2)]: said page including data dynamically retrieved from one or more data sources	The generated web page includes data dynamically retrieved from one or more data sources by means of the PL/SQL procedures, which access an SQL database. "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." MAC003054.	

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
16(a): The computer-implemented method in claim 15 wherein said step of transferring said request includes the steps of:	OIS is computer-implemented	Dr. Shamos did not prove that Reference 153 meets the preamble.
16(b): transferring said request from said HTTP-compliant device to a dispatcher, and	The web agent in OracleServer acts as a dispatcher. "The Oracle Web Agent is a CGI program that the Web Listener executes when a request is received for a dynamic document. To connect to different servers, or different schemas on the same server, multiple Web Agent services may be configured on a single Oracle WebServer." MAC003054. Thus the Web Agent makes an informed choice of which Web Agent service is to be invoked to handle the request. "When a request from a Web browser comes in, the Web Listener will extract the service name that is embedded in the URL and find out which parameters to use by reading the owa.cfg file." MAC003148. This is configuration file dispatching.	As discussed in the section below, Dr. Shamos failed to establish that Reference 153 has a "dispatcher" because, as shown below, Reference 153 does not disclose "dispatching" under either Oracle's or epicRealm's proposed construction. As explained below, the Web Agent cannot be a dispatcher because the Web Agent has no choice of which Oracle7 Server to connect to. (Ex. 53, Montinola Dep. at 81). The Web agent merely follows the instructions contained in the URL in conjunction with a configuration file. (Ex. 53, Montinola Dep. at 61-62, 80-81). Furthermore, Dr. Shamos failed to identify any dynamic information that the Web Agent maintains about page servers indicating which page server can more efficiently process the request.
16(c): dispatching said request to said page server	Oracle WebServer 1.0 also performs SQL*Net dispatching. Page 1-3. "To connect to the Oracle7 Server, the Web Agent needs the following information to be specified in the Web Agent service: • username • password • ORACLE_HOME • ORACLE_SID (for local databases only) • SQL*Net V2 Service Name or Connect String (for remote databases only). The Administration Utility allows the administrator to display, create, modify or delete a Web Agent service. With the Web Agent Creation form, you do not need to modify the configuration	Reference 153 does not disclose "dispatching" under either Oracle's or epicRealm's proposed construction. First, Dr. Shamos contended that his "dispatcher" of Reference 153 (Web Agent) makes an informed selection of which "page server" to use. However, Dr. Shamos did not distinguish between epicRealm's construction and Oracle's construction. EpicRealm's proposed construction for "dispatching" is "examining a request to make an informed selection of which page server should process the request based on dynamic

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
	file for the Oracle Web Agent service (owa.cfg) directly." MAC003148-3150.	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." Dr. Shamos failed to identify any dynamic information maintained about page servers indicating which page server can more efficiently process the request. The configuration file does not maintain dynamic information because the configuration file contains static information and is manually changed by a programmer, not automatically changed. I address Dr. Shamos's contention regarding "Configuration File Dispatching" earlier in my report. Furthermore, the configuration file does not maintain information about which page server can more efficiently process a request. The configuration file exists in storage memory of the server and specifies database connection information for the *only* database to process the request. The configuration file does not distinguish between the relative efficiencies of page servers. Thus, Reference 153 does not disclose "dispatching" under epicRealm's proposed construction. Similarly, Dr. Shamos failed to demonstrate that Reference 153 discloses "dispatching" under Oracle's construction. At a minimum, Dr. Shamos did not distinguish his analysis under epicRealm's and Oracle's proposed construction for "dispatching." Therefore, for the reasons set forth above, Reference 153 failed to disclose "dispatching." In addition, Oracle's construction requires that the dispatcher make an "informed selection of which page server should process the request...." In Reference 153, there is not

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
		an "informed selection" because the request is routed to the database that is specified in the configuration file based on the URL. In other words, the dispatcher is not making a selection from the available "page servers." Oracle's corporate witness on Oracle WebServer 1.0 testified that the Web Agent has no choice of which Oracle7 Server to connect to. (Ex. 53, Montinola Dep. at 81). The Web agent merely follows the instructions contained in the URL in conjunction with a configuration file. (Ex. 53, Montinola Dep. at 61-62, 80-81). The Web Agent also has no real-time status information about the Oracle7 Server. (Ex. 53, Montinola Dep. at 81). Oracle, who considered itself a market leader who released state-of-the-art products, readily acknowledged that prior to its release of Oracle Webserver 2.0, Oracle was not aware of any web servers that used load balancing or any programs that generated dynamic web pages that would have used any type of load balancing. (Ex. 53, Montinola Dep. at 221). Thus, Reference 153 does not disclose "dispatching" under Oracle's proposed construction.
17: The computer-implemented method in claim 15 wherein said step of processing said request includes the step of identifying said one or more data sources from which to retrieve said data	"After connecting to the database the Web Agent invokes the appropriate PL/SQL procedure, whose name is obtained by parsing the PATH_INFO environment variable." MAC003146. The data sources to be used are identified by examining the PATH_INFO environment variable.	Not asserted.
18: The computer-implemented method in claim 15 wherein said step of	"The Oracle WebServer is an HTTP server with a tightly integrated Oracle7 Server that enables the creation of dynamic HTML documents from data	Not asserted.

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
dynamically generating said Web page includes the step of dynamically retrieving said data from said one or more data sources	stored in an Oracle database. When the data changes, these HTML documents are updated automatically….This approach supplements the presentation of static, or unchanging, data which is found on most sites today, with the dynamic real-time data present in business systems based on the Oracle 7 Server." p. 1-2 (MAC003052). 5. The PL/SQL procedure executes, generating an HTML document. With the help of the Developer's Toolkit, the PL/SQL procedure extracts data from the Oracle7 database and generates an HTML document in a PL/SQL table." (Oracle, p. 5-5).	
19: The computer-implemented method in claim 15 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more data sources	Connection caching is used in SQL*Net, which may optionally be used with OWS 1.0. ORCL01806717	Not asserted.
20: The computer-implemented method in claim 15 wherein said step of processing said request includes the step of logging into said one or more data sources	"In order to connect to an Oracle 7 Server, the Web Agent requires certain information, such as which server to connect to and what username and password to use." (MAC003146). The username and password are used to log in to the data source. "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	Not asserted.

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
	applications that unify data from several different servers, while controlling exactly what information a Web client can access." (p. 1-4 (MAC 3054; ORCL 23). "When the Oracle Web Agent logs into the Oracle7 Server, it starts a PL/SQL procedure that has been created by the user to generate an HTML page as its output." (p. 1-4 (MAC 3054; ORCL 23).)	
21: The computer-implemented method in claim 15 wherein said step of dynamically generating said page includes the step of maintaining a page cache containing said page	Pages MAC003097-98, MAC003223-24 and MAC003255-56 explain how to set up file caching, which "decreases the access time required for commonly-used files." "The Oracle Web Listener allows a configurable set of commonly accessed files to be cached in memory. This provides very good performance when these files are accessed by a client." MAC003086	Not asserted.
22: The computer-implemented method in claim 15 wherein said page server includes tag-based text templates for configuring said page	OIS uses HTML extension templates, which are tag-based. See '554 7	Not asserted.
23: The computer-implemented method in claim 22 wherein said step of processing said request further includes the step of inserting said dynamically retrieved data from said one or more data sources into said tag-based text templates	See '554 8.	Not asserted.
24: The computer-	Section 5 (e.g., page 5-5, MAC003147) describes the	Not asserted.

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
implemented method in claim 22 wherein at least one of said tag-based text templates drives a format of the data dynamically retrieved from said one or more data sources in response to said request	use of tag-based text templates in PL/SQL procedures to drive the format of the data in creating the generated page. "The Oracle WebServer Developer's Toolkit is a set of procedures that help the customer create programs to generate dynamic documents that are easily presented in HTML." p. 1-2 (MAC003052). HTML determines the format of a web page	
25: The computer-implemented method in claim 22 wherein said tag-based text templates include HTML templates	See '554 8.	Not asserted.
26: The computer-implemented method in claim 15 wherein said step of processing said request further includes the step of dynamically updating data at said one or more data sources	Dynamically updating the database is described on page 4-9 (MAC003113). "Which SQL statements determine the actual data content? Chiefly, three – INSERT, UPDATE, and DELETE. INSERT places rows in a table, UPDATE changes the values they contain, and DELETE removes them." p. 4-8 (MAC003112)	Not asserted.
27: The computer-implemented method in claim 15 wherein said step of processing said request further includes the step of processing an object handling extension	N/A	Not asserted.
28: The computer-implemented method in claim 27 wherein said object handling extension is an OLE extension	N/A	Not asserted.

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
29(a): A computer-implemented method comprising the steps of:	OIS is computer-implemented	Not asserted.
29(b): transferring a request from an HTTP-compliant device to a dispatcher	The web agent in OracleServer acts as a dispatcher. "The Oracle Web Agent is a CGI program that the Web Listener executes when a request is received for a dynamic document. To connect to different servers, or different schemas on the same server, multiple Web Agent services may be configured on a single Oracle WebServer." MAC003054. Thus the Web Agent makes an informed choice of which Web Agent service is to be invoked to handle the request. "When a request from a Web browser comes in, the Web Listener will extract the service name that is embedded in the URL and find out which parameters to use by reading the owa.cfg file." MAC003148. This is configuration file dispatching.	Not asserted.
29(c): maintaining dynamic information regarding data sources a given page server may access	Page 4-5 (MAC003109) describes how the page server (Oracle7 Server) can access multiple databases and teaches a facility known as SQL*Net, which communicates with components of a distributed database. It is inherent in this structure that the page server maintains information on which data sources it may access. "If you have a full license for the Oracle7 Server, you can use it to create distributed databases. These are multiple Oracle7 databases – located on different machines and possibly separated by considerable geographical distance – that work together to maintain a single consistent body of data that looks and behaves as though drawn from a single source. In this situation, you connect to one database, which is called the local database, and that database	Not asserted.

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
	interacts with other databases, called remote databases, as needed." p. 4-5 (MAC003109)	
29(d)(1): dispatching said request to an appropriate page server based on said request and based on said dynamic information	Oracle WebServer 1.0 also performs SQL*Net dispatching. Page 1-3. "To connect to the Oracle7 Server, the Web Agent needs the following information to be specified in the Web Agent service: • username • password • ORACLE_HOME • ORACLE_SID (for local databases only) • SQL*Net V2 Service Name or Connect String (for remote databases only). The Administration Utility allows the administrator to display, create, modify or delete a Web Agent service. With the Web Agent Creation form, you do not need to modify the configuration file for the Oracle Web Agent service (owa.cfg) directly." MAC003148-3150.	Not asserted.
29(d)(2): said page server receiving said request and	The diagram on page MAC003053 shows that the request is received by the Oracle7 Server (page server).	Not asserted.
29(d)(3): releasing said HTTP-compliant device to process other requests	"For maximum performance the Web Listener is designed to run as an asynchronous engine … providing high performance under a heavy load." p. 3-2. Here "asynchronous" means that it functions independently of other tasks, which are being performed simultaneously. For the web listener (web server) to run asynchronously, it must be released to respond to other requests after passing off a request to a page server. Further, OIS passes requests using HTTP, so TCP releasing is performed. Page 1-3 shows SQL*Net running over a TCP/IP connection. The "releasing limitation" is met under the TCP	Not asserted.

"Oracle WebServer User's Guide Release 1.0 Part Number A349862" (153)

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
	theory of releasing. Furthermore, the Oracle7 server binds to a TCP port and therefore the releasing limitation is met under the SBL theory of releasing. MAC003053	
29(e): processing said request, said processing being performed by said page server while said HTTP-compliant device concurrently processes said other requests, and	The Oracle7 server (page server) processes the dynamic page generation request and returns the generated page to the web agent. This process occurs simultaneously with the web listener (web server) processing other requests because the Web Listener operates asynchronously. "The Oracle Web Listener can handle a large number of simultaneous requests, and has advanced features which use system resources more efficiently than other HTTP servers available on the market." MAC003053. Also, OWS 1.0 allows for instantiation of multiple web listeners, allowing for concurrent processing of requests.	Not asserted.
29(f)(1): dynamically generating a Web page in response to said request	See '554 1c	Not asserted.
29(f)(2): said page including data dynamically retrieved from one or more data sources	The generated web page includes data dynamically retrieved from one or more data sources by means of the PL/SQL procedures, which access an SQL database. "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." MAC003054.	Not asserted.

Exhibit 8

Popp et al. U.S. Patent No. 6,249,291 (195)

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
1(a): A computer-implemented method for managing a dynamic Web page generation request to a Web server, said computer-implemented method comprising the steps of:	Some or all of a Web page can be generated dynamically using input received in a returned page, generated at runtime, or retrieved from an external data source (e.g., database or electronic mail system). Abstract	Dr. Shamos did not prove that Reference 195 (Ex. 46) meets the preamble.
1(b)(1): routing said request from said Web server to a page server	Page server = application server 316. "A client request is forwarded to application server 316 via HTTP server 314." 8:20-21	Under Oracle's proposed construction, the "page server" is "page generating software, running on a processor separate from that of the Web server, that generates dynamic Web pages." Dr. Shamos failed to prove that Reference 195 discloses a page server under Oracle's proposed construction because he failed to identify a "page server" that executes on a processor separate from the processor of the Web Server.
1(b)(2): said page server receiving said request and	The request must be received to be processed.	
1(b)(3): releasing said Web server to process other requests	TCP releasing	Dr. Shamos contended that "TCP Releasing" disclosed by Reference 195 meets this limitation under either party's claim construction. I agree.
1(b)(4): wherein said routing step further includes the steps of intercepting said request at said Web server	Intercepting is what sends the request to the CGI program 322 instead of the web server	
1(b)(5): routing said request from said Web server to a dispatcher, and	CGI program 322 is the dispatcher. "CGI program 322 passes the shopper's input contained in the "Welcome" page to the application 320 running on application server 316 via corporate network 312." 9:41-44. "The application name identifies the application that services the client request. If an application is specified, CGIMessenger 210 transmits the client request and corresponding information transmitted from HTTP Server 206 to application 214. Application 214 can execute on the same or	As discussed in the section below, Dr. Shamos failed to establish that Reference 195 has a "dispatcher" because Reference 195 does not disclose "dispatching" under either Oracle's or epicRealm's proposed construction.

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
	different server as CGIMessenger 210 and/or HTTP Server 206, for example. Application 214 executes an interaction flow to satisfy the user request. Application 214 can access an external data source such as database 224. Database 224 can be resident on the same server as application 214. Alternatively, database 224 can be resident on a separate server (e.g., a separate database server)." 7:24-35	
1(b)(6): dispatching said request to said page server	See '554 1b5. " For more sophisticated requests, the present invention can execute logic such as CGIMessenger 210 and application 214, for example, to process a client request. The application preferably includes an object class hierarchy that can be implemented in any object-oriented language such as objective C, SmallTalk or JAVA. Objects (e.g., Web page definitional objects and control objects) dynamically generate a Web page and manage client interaction via the Internet and WWW." 7:50-58.	Reference 195 does not disclose "dispatching" under either party's construction. EpicRealm's proposed construction for "dispatching" is "examining a request to make an informed selection of which page server should process the request based on dynamic information maintained about page servers, the dynamic information indicating which page server can more efficiently process the request, and sending the request to the selected page server." Reference 195 neither discloses maintaining "dynamic information indicating which page server can more efficiently process the request" nor "making an informed selection of which page server should process the request" based on that dynamic information. Reference 195 discloses forwarding requests to the application specified by the application name included in the request. This information is not "dynamic information that is maintained" because the information is part of the request and is discarded after use. In addition, this information does not specify which page server can more efficiently process a request because it indicates where the request must be routed -- there is no evaluation of the efficiency of a "page server." Further, there is no informed selection of "which page

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
		server should process the request" because Reference 195 does not disclose distinguishing between multiple "page servers" that can service the request. Similarly, Dr. Shamos failed to demonstrate that Reference 195 discloses "dispatching" under Oracle's construction. At a minimum, Dr. Shamos did not distinguish his analysis under epicRealm's and Oracle's proposed construction for "dispatching." Therefore, for the reasons set forth above, Reference 195 fails to disclose "dispatching" under Oracle's construction. In addition, Reference 195 does not disclose "dispatching" under Oracle's construction because Reference 195 does not disclose making an informed selection about which page server should process a request. The CGI program sends a request to the only server that can process the request. Reference 195 only discloses routing to a singular application.
1(c): processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests, and	The web server and page server are different machines, so concurrent processing occurs inherently. "If an application is specified, CGIMessenger 210 transmits the client request and corresponding information transmitted from HTTP Server 206 to application 214. Application 214 can execute on the same or different server as CGIMessenger 210 and/or HTTP Server 206, for example. Application 214 executes an interaction flow to satisfy the user request. Application 214 can access an external data source such as database 224. Database 224 can be resident on the same server as application 214. Alternatively, database 224 can be resident on a separate server (e.g., a separate database server)." 7:25-35.	There is no express disclosure of concurrent processing in Reference 195. Concurrent processing requires that that "Web server" is capable of processing a request while the "page server" processes a request. Reference 195 does not expressly disclose such capabilities.

Popp et al. U.S. Patent No. 6,249,291 (195)

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
1(d)(1): dynamically generating a Web page in response to said request	The application 214 returns "a Web page (e.g., an HTML Web page) that is dynamically generated using complex queries (or other data retrieval mechanisms) to retrieve data and dynamically generate an HTML page using complex logic." 7:44-58.	
1(d)(2): said Web page including data dynamically retrieved from one or more data sources.	The contents of the Web page can contain data retrieved from an external data source (e.g., such as database server 318). 8:44-46	
2: The computer-implemented method in claim 1 wherein said step of processing said request includes the step of identifying said one or more data sources from which to retrieve said data.	CGI server 906 identifies the Web Object server 912 based on the information contained in the request, for example. 26:3-4	Reference 195 cannot anticipate this claim because it does not anticipate the claim from which this claim depends.
3: The computer-implemented method in claim 2 wherein said step of dynamically generating said Web page includes the step of dynamically retrieving said data from said one or more data sources.	For example, the present invention can be used to access a Web page (e.g., an HTML Web page) that is dynamically generated using complex queries (or other data retrieval mechanisms) to retrieve data and dynamically generate an HTML page using complex logic. 7:45-49	Reference 195 cannot anticipate this claim because it does not anticipate the claim from which this claim depends.
4: The computer-implemented method in claim 3 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more data sources.	State information can be retained to allow the internal application to allow the application to maintain open sessions with multiple users capable of processing multiple transactions in any order. 554 8:28-31. "For example, a connection to a database can be used at the global level by multiple sessions and applications." 13:38-40.	Reference 195 cannot anticipate this claim because it does not anticipate the claim from which this claim depends. Dr. Shamos failed to demonstrate how Reference 195 discloses maintaining a connection cache. He does not identify a cache of reusable connections to a database.
5: The computer-implemented	A digital signature can be used as a key to perform	Reference 195 cannot anticipate this claim because it

Popp et al. U.S. Patent No. 6,249,291 (195)

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
method in claim 3 wherein said step of processing said request includes the step of logging into said one or more data sources.	verification. For example, the digital signature can be used to verify that a request is a valid request generated by a valid client. The digital signature can further be used to verify that the client is authorized to access the data of the specified session. 27:12-17	does not anticipate the claim from which this claim depends.
6: The computer-implemented method in claim 3 wherein said step of dynamically generating said Web page includes the step of maintaining a page cache containing said Web page.	N/A	Not asserted.
7: The computer-implemented method in claim 3 wherein said page server includes custom HTML extension templates for configuring said Web page.	The present invention provides an extension to HTML that is used on the HTTP server side. The HTML extension is filtered out before a Web page is sent to a client browser. It is used to interpret an HTML template and to render an HTML document before is transmitted to the client browser 15:45-50	Reference 195 cannot anticipate this claim because it does not anticipate the claim from which this claim depends.
8: The computer-implemented method in claim 7 wherein said step of processing said request further includes the step of inserting said dynamically retrieved data from said one or more data sources into said custom HTML extension templates.	A stringcontrol object instance instantiates an NSWHString static HTML element and inserts it in the template. 18:57-59	Reference 195 cannot anticipate this claim because it does not anticipate the claim from which this claim depends.
9(a): A networked system for managing a dynamic Web page generation request, said system comprising	Some or all of a Web page can be generated dynamically using input received in a returned page, generated at runtime, or retrieved from an external data source (e.g., database or electronic mail system). Abstract. Networked: "The present invention provides the ability to access an application using any	Dr. Shamos did not prove that Reference 195 meets the preamble.

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
	interface. For example, a client can access the same application via the Internet using a system running Windows, MAC/OS, Sun.OS, NextStep, etc. Referring to FIG. 2, client 202 is resident on a corporate network 200. Corporate network 200 is a local area network comprised of personal computers such as client 202, for example." 6:32-40	
9(b): one or more data sources	The contents of the Web page can contain data retrieved from an external data source (e.g., such as database server 318). 8:44-46	
9(c): a page server having a processing means	Page server = application server 316. "A client request is forwarded to application server 316 via HTTP server 314." 8:20-21	Under Oracle's proposed construction, the "page server" is "page generating software, running on a processor separate from that of the second computer system, that generates dynamic Web pages." Dr. Shamos failed to prove that Reference 195 discloses a page server under Oracle's proposed construction because he failed to identify a "page server" that executes on a processor separate from the processor of the second computer system. Dr. Shamos failed to identify the specified function and the corresponding structure, material or acts for the "processing means" as required by 35 U.S.C. § 112 ¶ 6. Furthermore, Dr. Shamos failed to demonstrate that Reference 195 discloses the specified function and corresponding structure, material or acts for the "processing means" or equivalents thereof.
9(d): a first computer system including means for generating said request, and	First computer system is the client. "Requests submitted by clients 302, 304, 306, and 308 to HTTP server 314 are transmitted to HTTP server 314 via WWW 310." 8:17-19.	Dr. Shamos failed to identify the specified function and the corresponding structure, material or acts for the "means for generating said request" as required by 35 U.S.C. § 112 ¶ 6. Furthermore, Dr. Shamos failed to demonstrate that Reference 194 that discloses the specified function and corresponding structure,

Popp et al. U.S. Patent No. 6,249,291 (195)

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
		material or acts for the "means for generating said request" or equivalents thereof.
9(e)(1): a second computer system including	The second computer system is box 208 in Fig. 2 and everything to its right	
9(e)(2): means for receiving said request from said first computer	Page server = application server 316. "A client request is forwarded to application server 316 via HTTP server 314." 8:20-21	Dr. Shamos failed to identify the specified function and the corresponding structure, material or acts for the "means for receiving" as required by 35 U.S.C. § 112 ¶ 6. Furthermore, Dr. Shamos failed to demonstrate that Reference 194 discloses the specified function and corresponding structure, material or acts for the "means for receiving" or equivalents thereof.
9(e)(3): said second computer system also including a router, said router routing said request from said second computer system to said page server	Page server = application server 316. "A client request is forwarded to application server 316 via HTTP server 314." 8:20-21	
9(e)(4): wherein said routing further includes intercepting said request at said second computer	Intercepting is what sends the request to the CGI program 322 instead of the web server	
9(e)(5): routing said request from said second computer to a dispatcher, and	CGI program 322 is the dispatcher. "CGI program 322 passes the shopper's input contained in the "Welcome" page to the application 320 running on application server 316 via corporate network 312." 9:41-44. "The application name identifies the application that services the client request. If an application is specified, CGIMessenger 210 transmits the client request and corresponding information transmitted from HTTP Server 206 to application 214. Application 214 can execute on the same or different server as CGIMessenger 210 and/or HTTP Server 206, for example. Application 214 executes an interaction flow to satisfy the user request.	As discussed in the section below, Dr. Shamos failed to establish that Reference 195 has a "dispatcher" because Reference 195 does not disclose "dispatching" under either Oracle's or epicRealm's proposed construction.

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
	Application 214 can access an external data source such as database 224. Database 224 can be resident on the same server as application 214. Alternatively, database 224 can be resident on a separate server (e.g., a separate database server)." 7:24-35	
9(e)(6): dispatching said request to said page server	See '554 1b5. " For more sophisticated requests, the present invention can execute logic such as CGIMessenger 210 and application 214, for example, to process a client request. The application preferably includes an object class hierarchy that can be implemented in any object-oriented language such as objective C, SmallTalk or JAVA. Objects (e.g., Web page definitional objects and control objects) dynamically generate a Web page and manage client interaction via the Internet and WWW." 7:50-58.	Reference 195 does not disclose "dispatching" under either party's construction. EpicRealm's proposed construction for "dispatching" is "examining a request to make an informed selection of which page server should process the request based on dynamic information maintained about page servers, the dynamic information indicating which page server can more efficiently process the request, and sending the request to the selected page server." Reference 195 neither discloses maintaining "dynamic information indicating which page server can more efficiently process the request" nor "making an informed selection of which page server should process the request" based on that dynamic information. Reference 195 discloses forwarding requests to the application specified by the application name included in the request. This information is not "dynamic information that is maintained" because the information is part of the request and is discarded after use. In addition, this information does not specify which page server can more efficiently process a request because it indicates where the request must be routed -- there is no evaluation of the efficiency of a "page server." Further, there is no informed selection of "which page server should process the request" because Reference 195 does not disclose distinguishing between multiple "page servers" that can service the request.

Popp et al. U.S. Patent No. 6,249,291 (195)

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
		Similarly, Dr. Shamos failed to demonstrate that Reference 195 discloses "dispatching" under Oracle's construction. At a minimum, Dr. Shamos did not distinguish his analysis under epicRealm's and Oracle's proposed construction for "dispatching." Therefore, for the reasons set forth above, Reference 195 fails to disclose "dispatching" under Oracle's construction. In addition, Reference 195 does not disclose "dispatching" under Oracle's construction because Reference 195 does not disclose making an informed selection about which page server should process a request. The CGI program sends a request to the only server that can process the request. Reference 195 only discloses routing to a singular application.
9(e)(7): said page server receiving said request and	The request must be received to be processed	
9(e)(8): releasing said second computer system to process other requests	TCP releasing	Dr. Shamos contended that "TCP Releasing" disclosed by Reference 195 meets this limitation under either party's claim construction. I agree.
9(e)(9): said page server processing means processing said request and dynamically generating a Web page in response to said request	The application 214 returns "a Web page (e.g., an HTML Web page) that is dynamically generated using complex queries (or other data retrieval mechanisms) to retrieve data and dynamically generate an HTML page using complex logic." 7:44-58.	
9(e)(10): said Web page including data dynamically retrieved from said one or more data sources	The contents of the Web page can contain data retrieved from an external data source (e.g., such as database server 318). 8:44-46	
10(a): The networked system in claim 9 wherein said router in said second computer	Some or all of a Web page can be generated dynamically using input received in a returned page, generated at runtime, or retrieved from an external	Reference 195 cannot anticipate this claim because it does not anticipate the claim from which this claim depends.

Popp et al. U.S. Patent No. 6,249,291 (195)

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
system includes:	data source (e.g., database or electronic mail system). Abstract. Networked: "The present invention provides the ability to access an application using any interface. For example, a client can access the same application via the Internet using a system running Windows, MAC/OS, Sun.OS, NextStep, etc. Referring to FIG. 2, client 202 is resident on a corporate network 200. Corporate network 200 is a local area network comprised of personal computers such as client 202, for example." 6:32-40	
10(b): an interceptor intercepting said request at said second computer system and routing said request, and	Intercepting is what sends the request to the CGI program 322 instead of the web server	Reference 195 cannot anticipate this claim because it does not anticipate the claim from which this claim depends. Dr. Shamos did not identify an "interceptor" that intercepts the request at the second computer system in Reference 195.
10(c): a dispatcher receiving said routed request from said interceptor and dispatching said request to said page server	See '554 1b5. " For more sophisticated requests, the present invention can execute logic such as CGIMessenger 210 and application 214, for example, to process a client request. The application preferably includes an object class hierarchy that can be implemented in any object-oriented language such as objective C, SmallTalk or JAVA. Objects (e.g., Web page definitional objects and control objects) dynamically generate a Web page and manage client interaction via the Internet and WWW." 7:50-58.	Reference 195 cannot anticipate this claim because it does not anticipate the claim from which this claim depends. Reference 195 does not disclose "dispatching" under either party's construction. EpicRealm's proposed construction for "dispatching" is "examining a request to make an informed selection of which page server should process the request based on dynamic information maintained about page servers, the dynamic information indicating which page server can more efficiently process the request, and sending the request to the selected page server." Reference 195 neither discloses maintaining "dynamic information indicating which page server can more efficiently

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
		process the request" nor "making an informed selection of which page server should process the request" based on that dynamic information. Reference 195 discloses forwarding requests to the application specified by the application name included in the request. This information is not "dynamic information that is maintained" because the information is part of the request and is discarded after use. In addition, this information does not specify which page server can more efficiently process a request because it indicates where the request must be routed -- there is no evaluation of the efficiency of a "page server." Further, there is no informed selection of "which page server should process the request" because Reference 195 does not disclose distinguishing between multiple "page servers" that can service the request. Similarly, Dr. Shamos failed to demonstrate that Reference 195 discloses "dispatching" under Oracle's construction. At a minimum, Dr. Shamos did not distinguish his analysis under epicRealm's and Oracle's proposed construction for "dispatching." Therefore, for the reasons set forth above, Reference 195 fails to disclose "dispatching" under Oracle's construction. In addition, Reference 195 does not disclose "dispatching" under Oracle's construction because Reference 195 does not disclose making an informed selection about which page server should process a request. The CGI program sends a request to the only server that can process the request. Reference 195 only discloses routing to a singular application.
11(a): A machine readable medium having stored thereon	Because the system is a programmed system, it must have a medium on which instructions are stored	Dr. Shamos did not prove that Reference 195 meets the preamble.

Popp et al. U.S. Patent No. 6,249,291 (195)

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
data representing sequences of instructions, which when executed by a computer system, cause said computer system to perform the steps of:		
11(b)(1): routing a dynamic Web page generation request from a Web server to a page server	Page server = application server 316. "A client request is forwarded to application server 316 via HTTP server 314." 8:20-21	Under Oracle's proposed construction, the "page server" is "page generating software, running on a processor separate from that of the Web server, that generates dynamic Web pages." Dr. Shamos failed to prove that Reference 195 discloses a page server under Oracle's proposed construction because he failed to identify a "page server" that executes on a processor separate from the processor of the Web server.
11(b)(2): said page server receiving said request and	The request must be received to be processed	
11(b)(3): releasing said Web server to process other requests	TCP releasing	Dr. Shamos contended that "TCP Releasing" disclosed by Reference 195 meets this limitation under either party's claim construction. I agree.
11(b)(4): wherein said routing step further includes the steps of intercepting said request at said Web server	Intercepting is what sends the request to the CGI program 322 instead of the web server	
11(b)(5): routing said request from said Web server to a dispatcher, and	CGI program 322 is the dispatcher. "CGI program 322 passes the shopper's input contained in the "Welcome" page to the application 320 running on application server 316 via corporate network 312." 9:41-44. "The application name identifies the application that services the client request. If an application is specified, CGIMessenger 210 transmits the client request and corresponding information transmitted from HTTP Server 206 to application 214. Application 214 can execute on the same or	Dr. Shamos failed to establish that Reference 195 has a "dispatcher" because Reference 195 does not disclose "dispatching" under either Oracle's or epicRealm's proposed construction.

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
	different server as CGIMessenger 210 and/or HTTP Server 206, for example. Application 214 executes an interaction flow to satisfy the user request. Application 214 can access an external data source such as database 224. Database 224 can be resident on the same server as application 214. Alternatively, database 224 can be resident on a separate server (e.g., a separate database server)." 7:24-35	
11(b)(6): dispatching said request to said page server	See '554 1b5. " For more sophisticated requests, the present invention can execute logic such as CGIMessenger 210 and application 214, for example, to process a client request. The application preferably includes an object class hierarchy that can be implemented in any object-oriented language such as objective C, SmallTalk or JAVA. Objects (e.g., Web page definitional objects and control objects) dynamically generate a Web page and manage client interaction via the Internet and WWW." 7:50-58.	Reference 195 does not disclose "dispatching" under either party's construction. EpicRealm's proposed construction for "dispatching" is "examining a request to make an informed selection of which page server should process the request based on dynamic information maintained about page servers, the dynamic information indicating which page server can more efficiently process the request, and sending the request to the selected page server." Reference 195 neither discloses maintaining "dynamic information indicating which page server can more efficiently process the request" nor "making an informed selection of which page server should process the request" based on that dynamic information. Reference 195 discloses forwarding requests to the application specified by the application name included in the request. This information is not "dynamic information that is maintained" because the information is part of the request and is discarded after use. In addition, this information does not specify which page server can more efficiently process a request because it indicates where the request must be routed -- there is no evaluation of the efficiency of a "page server." Further, there is no informed selection of "which page

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
		server should process the request" because Reference 195 does not disclose distinguishing between multiple "page servers" that can service the request. Similarly, Dr. Shamos failed to demonstrate that Reference 195 discloses "dispatching" under Oracle's construction. At a minimum, Dr. Shamos did not distinguish his analysis under epicRealm's and Oracle's proposed construction for "dispatching." Therefore, for the reasons set forth above, Reference 195 fails to disclose "dispatching" under Oracle's construction. In addition, Reference 195 does not disclose "dispatching" under Oracle's construction because Reference 195 does not disclose making an informed selection about which page server should process a request. The CGI program sends a request to the only server that can process the request. Reference 195 only discloses routing to a singular application.
11(c): processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests, and	The web server and page server are different machines, so concurrent processing occurs inherently. "If an application is specified, CGIMessenger 210 transmits the client request and corresponding information transmitted from HTTP Server 206 to application 214. Application 214 can execute on the same or different server as CGIMessenger 210 and/or HTTP Server 206, for example. Application 214 executes an interaction flow to satisfy the user request. Application 214 can access an external data source such as database 224. Database 224 can be resident on the same server as application 214. Alternatively, database 224 can be resident on a separate server (e.g., a separate database server)." 7:25-35.	There is no express disclosure of concurrent processing in Reference 195. Concurrent processing requires that that "Web server" is capable of processing a request while the "page server" processes a request. Reference 195 does not expressly disclose such capabilities.

Popp et al. U.S. Patent No. 6,249,291 (195)

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
11(d): dynamically generating a Web page, said Web page including data retrieved from one or more data sources	The application 214 returns "a Web page (e.g., an HTML Web page) that is dynamically generated using complex queries (or other data retrieval mechanisms) to retrieve data and dynamically generate an HTML page using complex logic." 7:44-58.	Dr. Shamos only contended that Reference 195 discloses "dynamically generating a Web page in response to said request." Dr. Shamos failed to contend that Reference 195 discloses "dynamically generating a Web page, said Web page including data retrieved from one or more data sources."

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
2[1(a)]: A computer-implemented method for managing a dynamic Web page generation request to a Web server, said computer-implemented method comprising the steps of:	Some or all of a Web page can be generated dynamically using input received in a returned page, generated at runtime, or retrieved from an external data source (e.g., database or electronic mail system). Abstract	Dr. Shamos did not prove that Reference 195 meets the preamble.
2[1(b)(1)]: routing a request from a Web server to a page server	Page server = application server 316. "A client request is forwarded to application server 316 via HTTP server 314." 8:20-21	Under Oracle's proposed construction, the "page server" is "page generating software, running on a processor separate from that of the Web server, that generates dynamic Web pages." Dr. Shamos failed to prove that Reference 195 discloses a page server under Oracle's proposed construction because he failed to identify a "page server" that executes on a processor separate from the processor of the Web server.
2[1(b)(2)]: said page server receiving said request and	The request must be received to be processed	
2[1(b)(3)]: releasing said Web server to process other requests	TCP releasing	Dr. Shamos contended that "TCP Releasing" disclosed by Reference 195 meets this limitation under either party's claim construction. I agree.
2[1(c)(1)]: wherein said	Intercepting is what sends the request to the CGI	

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
routing step further includes the steps of: intercepting said request at said Web server and	program 322 instead of the web server	
2[1(c)(2)]: routing said request to said page server	Page server = application server 316. "A client request is forwarded to application server 316 via HTTP server 314." 8:20-21	
2[1(d)]: processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests, and	The web server and page server are different machines, so concurrent processing occurs inherently. "If an application is specified, CGIMessenger 210 transmits the client request and corresponding information transmitted from HTTP Server 206 to application 214. Application 214 can execute on the same or different server as CGIMessenger 210 and/or HTTP Server 206, for example. Application 214 executes an interaction flow to satisfy the user request. Application 214 can access an external data source such as database 224. Database 224 can be resident on the same server as application 214. Alternatively, database 224 can be resident on a separate server (e.g., a separate database server)." 7:25-35.	There is no express disclosure of concurrent processing in Reference 195. Concurrent processing requires that that "Web server" is capable of processing a request while the "page server" processes a request. Reference 195 does not expressly disclose such capabilities.
2[1(e)(1)]: dynamically generating a Web page in response to said request	The application 214 returns "a Web page (e.g., an HTML Web page) that is dynamically generated using complex queries (or other data retrieval mechanisms) to retrieve data and dynamically generate an HTML page using complex logic." 7:44-58.	
2[1(e)(2)]: said Web page including data dynamically retrieved from one or more data sources	The contents of the Web page can contain data retrieved from an external data source (e.g., such as database server 318). 8:44-46	
2(a): The computer-implemented method in claim	Some or all of a Web page can be generated dynamically using input received in a returned page,	Dr. Shamos did not prove that Reference 195 meets the preamble.

Popp et al. U.S. Patent No. 6,249,291 (195)

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
1 wherein said step of routing said request includes the steps of:	generated at runtime, or retrieved from an external data source (e.g., database or electronic mail system). Abstract	
2(b): routing said request from said Web server to a dispatcher, and	CGI program 322 is the dispatcher. "CGI program 322 passes the shopper's input contained in the "Welcome" page to the application 320 running on application server 316 via corporate network 312." 9:41-44. "The application name identifies the application that services the client request. If an application is specified, CGIMessenger 210 transmits the client request and corresponding information transmitted from HTTP Server 206 to application 214. Application 214 can execute on the same or different server as CGIMessenger 210 and/or HTTP Server 206, for example. Application 214 executes an interaction flow to satisfy the user request. Application 214 can access an external data source such as database 224. Database 224 can be resident on the same server as application 214. Alternatively, database 224 can be resident on a separate server (e.g., a separate database server)." 7:24-35	Dr. Shamos failed to establish that Reference 195 has a "dispatcher" because Reference 195 does not disclose "dispatching" under either Oracle's or epicRealm's proposed construction.
2(c): dispatching said request to said page server	See '554 1b5. " For more sophisticated requests, the present invention can execute logic such as CGIMessenger 210 and application 214, for example, to process a client request. The application preferably includes an object class hierarchy that can be implemented in any object-oriented language such as objective C, SmallTalk or JAVA. Objects (e.g., Web page definitional objects and control objects) dynamically generate a Web page and manage client interaction via the Internet and WWW." 7:50-58.	Reference 195 does not disclose "dispatching" under either party's construction. EpicRealm's proposed construction for "dispatching" is "examining a request to make an informed selection of which page server should process the request based on dynamic information maintained about page servers, the dynamic information indicating which page server can more efficiently process the request, and sending the request to the selected page server." Reference 195 neither discloses maintaining "dynamic information indicating which page server can more efficiently

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
		process the request" nor "making an informed selection of which page server should process the request" based on that dynamic information. Reference 195 discloses forwarding requests to the application specified by the application name included in the request. This information is not "dynamic information that is maintained" because the information is part of the request and is discarded after use. In addition, this information does not specify which page server can more efficiently process a request because it indicates where the request must be routed -- there is no evaluation of the efficiency of a "page server." Further, there is no informed selection of "which page server should process the request" because Reference 195 does not disclose distinguishing between multiple "page servers" that can service the request. Similarly, Dr. Shamos failed to demonstrate that Reference 195 discloses "dispatching" under Oracle's construction. At a minimum, Dr. Shamos did not distinguish his analysis under epicRealm's and Oracle's proposed construction for "dispatching." Therefore, for the reasons set forth above, Reference 195 fails to disclose "dispatching" under Oracle's construction. In addition, Reference 195 does not disclose "dispatching" under Oracle's construction because Reference 195 does not disclose making an informed selection about which page server should process a request. The CGI program sends a request to the only server that can process the request. Reference 195 only discloses routing to a singular application.
3: The computer-implemented method in claim 1 wherein said	CGI server 906 identifies the Web Object server 912 based on the information contained in the request, for	Not asserted.

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
step of processing said request includes the step of identifying said one or more data sources from which to retrieve said data	example. 26:3-4	
4: The computer-implemented method in claim 1 wherein said step of dynamically generating said Web page includes the step of dynamically retrieving said data from said one or more data sources	For example, the present invention can be used to access a Web page (e.g., an HTML Web page) that is dynamically generated using complex queries (or other data retrieval mechanisms) to retrieve data and dynamically generate an HTML page using complex logic. 7:45-49	Not asserted.
5: The computer-implemented method in claim 1 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more data sources	State information can be retained to allow the internal application to allow the application to maintain open sessions with multiple users capable of processing multiple transactions in any order. 554 8:28-31. "For example, a connection to a database can be used at the global level by multiple sessions and applications." 13:38-40.	Not asserted.
6: The computer-implemented method in claim 1 wherein said step of processing said request includes the step of logging into said one or more data sources	A digital signature can be used as a key to perform verification. For example, the digital signature can be used to verify that a request is a valid request generated by a valid client. The digital signature can further be used to verify that the client is authorized to access the data of the specified session. 27:12-17	Not asserted.
7: The computer-implemented method in claim 1 wherein said step of dynamically generating said Web page includes the step of maintaining a page cache containing said Web page	N/A	Not asserted.
8: The computer-implemented	Any reference with HTML templates (e.g. this one)	Not asserted.

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
method in claim 1 wherein said page server includes tag-based text templates for configuring said Web page	has tag-based text templates	
9: The computer-implemented method in claim 8 wherein said step of processing said request further includes the step of inserting said-dynamically retrieved data from said one or more data sources into said tag-based text templates	Each template contains a type, a Uniform Resource Locator (URL), and a list of byte count delimited attribute value pairs. p. 6. Also Fig. 3. A stringcontrol object instance instantiates an NSWHString static HTML element and inserts it in the template. 18:57-59	Not asserted.
10: The computer-implemented method in claim 8 wherein at least one of said tag-based text templates drives a format of the data dynamically retrieved from said one or more data sources in response to said request	The tag-based text templates are HTML templates, which drive a format of the data: "A language is used to define the format for a page to be displayed in the WWW. The language is called Hypertext Markup Language (HTML)." 2:26-28	Not asserted.
11: The computer-implemented method in claim 8 wherein said tag-based text templates include HTML templates	Any reference with HTML extension templates (e.g. this one) has HTML templates	Not asserted.
12: The computer-implemented method in claim 1 wherein said step of processing said request further includes the step of dynamically updating data at said one or more data sources	Claim 22: "computer readable program code configured to cause a computer to update said external data source with said object data"	Not asserted.
13: The computer-	The reference teaches generating dynamic pages	Not asserted.

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
implemented method in claim 1 wherein said step of processing said request further includes the step of processing an object handling extension	through objects: "All or some portion of the Web page is dynamically generated. Classes of object are associated with elements that defined a Web page (e.g., HTML elements). By objectifying Web pages it is possible to gain the advantages of object-oriented programming with Web pages." 4:5-9. This is accomplished through object handling extensions	
14: The computer-implemented method in claim 13 wherein said object handling extension is an OLE extension	N/A	Not asserted.
16[15(a)]: A computer-implemented method comprising the steps of:	The present invention can be implemented on a general purpose computer such as illustrated in FIG. 1. 5:56-57	Dr. Shamos did not prove that Reference 195 meets the preamble.
16[15(b)(1)]: transferring a request from an HTTP-compliant device to a page server	Page server = application server 316. "A client request is forwarded to application server 316 via HTTP server 314." 8:20-21	Under Oracle's proposed construction, the "page server" is "page generating software, running on a processor separate from that of the HTTP-compliant device, that generates dynamic Web pages." Dr. Shamos failed to prove that Reference 195 discloses a page server under Oracle's proposed construction because he failed to identify a "page server" that executes on a processor separate from the processor of the "HTTP-compliant device." Furthermore, Dr. Shamos's analysis for this limitation is flawed because he failed to identify a "HTTP-compliant device" nor does he factor either party's constructions for the claim terms in this limitation.
16[15(b)(2)]: said page server receiving said request and	The request must be received to be processed	
16[15(b)(3): releasing said HTTP-compliant device to	TCP releasing	Dr. Shamos contended that "TCP Releasing" disclosed by Reference 195 meets this limitation under either

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
process other requests		party's claim construction. I agree.
16[15(c)(1)]: wherein said transferring step further includes the steps of: intercepting said request at said HTTP-compliant device and	Intercepting is what sends the request to the CGI program 322 instead of the web server	
16[15(c)(2)]: transferring said request to said page server	Page server = application server 316. "A client request is forwarded to application server 316 via HTTP server 314." 8:20-21	
16[15(d)]: processing said request, said processing being performed by said page server while said HTTP-compliant device concurrently processes said other requests, and	The web server and page server are different machines, so concurrent processing occurs inherently. "If an application is specified, CGIMessenger 210 transmits the client request and corresponding information transmitted from HTTP Server 206 to application 214. Application 214 can execute on the same or different server as CGIMessenger 210 and/or HTTP Server 206, for example. Application 214 executes an interaction flow to satisfy the user request. Application 214 can access an external data source such as database 224. Database 224 can be resident on the same server as application 214. Alternatively, database 224 can be resident on a separate server (e.g., a separate database server)." 7:25-35.	There is no express disclosure of concurrent processing in Reference 195. Concurrent processing requires that that "Web server" is capable of processing a request while the "page server" processes a request. Reference 195 does not expressly disclose such capabilities.
16[15(e)(1)]: dynamically generating a page in response to said request	The application 214 returns "a Web page (e.g., an HTML Web page) that is dynamically generated using complex queries (or other data retrieval mechanisms) to retrieve data and dynamically generate an HTML page using complex logic." 7:44-58.	
16[15(e)(2)]: said page including data dynamically	The contents of the Web page can contain data retrieved from an external data source (e.g., such as	

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
retrieved from one or more data sources	database server 318). 8:44-46	
16(a): The computer-implemented method in claim 15 wherein said step of transferring said request includes the steps of:	The present invention can be implemented on a general purpose computer such as illustrated in FIG. 1. 5:56-57	Dr. Shamos did not prove that Reference 195 meets the preamble.
16(b): transferring said request from said HTTP-compliant device to a dispatcher, and	CGI program 322 is the dispatcher. "CGI program 322 passes the shopper's input contained in the "Welcome" page to the application 320 running on application server 316 via corporate network 312." 9:41-44. "The application name identifies the application that services the client request. If an application is specified, CGIMessenger 210 transmits the client request and corresponding information transmitted from HTTP Server 206 to application 214. Application 214 can execute on the same or different server as CGIMessenger 210 and/or HTTP Server 206, for example. Application 214 executes an interaction flow to satisfy the user request. Application 214 can access an external data source e such as database 224. Database 224 can be resident on the same server as application 214. Alternatively, database 224 can be resident on a separate server (e.g., a separate database server)." 7:24-35	Dr. Shamos failed to establish that Reference 195 has a "dispatcher" because Reference 195 does not disclose "dispatching" under either Oracle's or epicRealm's proposed construction.
16(c): dispatching said request to said page server	See '554 1b5. "For more sophisticated requests, the present invention can execute logic such as CGIMessenger 210 and application 214, for example, to process a client request. The application preferably includes an object class hierarchy that can be implemented in any object-oriented language such as objective C, SmallTalk or JAVA. Objects (e.g., Web page definitional objects and control objects)	Reference 195 does not disclose "dispatching" under either party's construction. EpicRealm's proposed construction for "dispatching" is "examining a request to make an informed selection of which page server should process the request based on dynamic information maintained about page servers, the dynamic information indicating which page server

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
	dynamically generate a Web page and manage client interaction via the Internet and WWW." 7:50-58.	can more efficiently process the request, and sending the request to the selected page server." Reference 195 neither discloses maintaining "dynamic information indicating which page server can more efficiently process the request" nor "making an informed selection of which page server should process the request" based on that dynamic information. Reference 195 discloses forwarding requests to the application specified by the application name included in the request. This information is not "dynamic information that is maintained" because the information is part of the request and is discarded after use. In addition, this information does not specify which page server can more efficiently process a request because it indicates where the request must be routed -- there is no evaluation of the efficiency of a "page server." Further, there is no informed selection of "which page server should process the request" because Reference 195 does not disclose distinguishing between multiple "page servers" that can service the request. Similarly, Dr. Shamos failed to demonstrate that Reference 195 discloses "dispatching" under Oracle's construction. At a minimum, Dr. Shamos did not distinguish his analysis under epicRealm's and Oracle's proposed construction for "dispatching." Therefore, for the reasons set forth above, Reference 195 fails to disclose "dispatching" under Oracle's construction. In addition, Reference 195 does not disclose "dispatching" under Oracle's construction because Reference 195 does not disclose making an informed selection about which page server should process a request. The CGI program sends a request to

Popp et al. U.S. Patent No. 6,249,291 (195)

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
		the only server that can process the request. Reference 195 only discloses routing to a singular application.
17: The computer-implemented method in claim 15 wherein said step of processing said request includes the step of identifying said one or more data sources from which to retrieve said data	CGI server 906 identifies the Web Object server 912 based on the information contained in the request, for example. 26:3-4	Not asserted.
18: The computer-implemented method in claim 15 wherein said step of dynamically generating said Web page includes the step of dynamically retrieving said data from said one or more data sources	For example, the present invention can be used to access a Web page (e.g., an HTML Web page) that is dynamically generated using complex queries (or other data retrieval mechanisms) to retrieve data and dynamically generate an HTML page using complex logic. 7:45-49	Not asserted.
19: The computer-implemented method in claim 15 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more data sources	State information can be retained to allow the internal application to allow the application to maintain open sessions with multiple users capable of processing multiple transactions in any order. 554 8:28-31. "For example, a connection to a database can be used at the global level by multiple sessions and applications." 13:38-40.	Not asserted.
20: The computer-implemented method in claim 15 wherein said step of processing said request includes the step of logging into said one or more data sources	A digital signature can be used as a key to perform verification. For example, the digital signature can be used to verify that a request is a valid request generated by a valid client. The digital signature can further be used to verify that the client is authorized to access the data of the specified session. 27:12-17	Not asserted.

Popp et al. U.S. Patent No. 6,249,291 (195)

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
21: The computer-implemented method in claim 15 wherein said step of dynamically generating said page includes the step of maintaining a page cache containing said page	N/A	Not asserted.
22: The computer-implemented method in claim 15 wherein said page server includes tag-based text templates for configuring said page	Any reference with HTML templates (e.g. this one) has tag-based text templates	Not asserted.
23: The computer-implemented method in claim 22 wherein said step of processing said request further includes the step of inserting said dynamically retrieved data from said one or more data sources into said tag-based text templates	Each template contains a type, a Uniform Resource Locator (URL), and a list of byte count delimited attribute value pairs. p. 6. Also Fig. 3. A stringcontrol object instance instantiates an NSWHString static HTML element and inserts it in the template. 18:57-59	Not asserted.
24: The computer-implemented method in claim 22 wherein at least one of said tag-based text templates drives a format of the data dynamically retrieved from said one or more data sources in response to said request	The tag-based text templates are HTML templates, which drive a format of the data: "A language is used to define the format for a page to be displayed in the WWW. The language is called Hypertext Markup Language (HTML)." 2:26-28	Not asserted.
25: The computer-implemented method in claim 22 wherein said tag-based text	Any reference with HTML extension templates (e.g. this one) has HTML templates	Not asserted.

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
templates include HTML templates		
26: The computer-implemented method in claim 15 wherein said step of processing said request further includes the step of dynamically updating data at said one or more data sources	Claim 22: "computer readable program code configured to cause a computer to update said external data source with said object data"	Not asserted.
27: The computer-implemented method in claim 15 wherein said step of processing said request further includes the step of processing an object handling extension	The reference teaches generating dynamic pages through objects: "All or some portion of the Web page is dynamically generated. Classes of object are associated with elements that defined a Web page (e.g., HTML elements). By objectifying Web pages it is possible to gain the advantages of object-oriented programming with Web pages." 4:5-9. This is accomplished through object handling extensions	Not asserted.
28: The computer-implemented method in claim 27 wherein said object handling extension is an OLE extension	N/A	Not asserted.
29(a): A computer-implemented method comprising the steps of:	The present invention can be implemented on a general purpose computer such as illustrated in FIG. 1. 5:56-57	Not asserted.
29(b): transferring a request from an HTTP-compliant device to a dispatcher	CGI program 322 is the dispatcher. "CGI program 322 passes the shopper's input contained in the "Welcome" page to the application 320 running on application server 316 via corporate network 312." 9:41-44. "The application name identifies the application that services the client request. If an application is specified, CGIMessenger 210 transmits the client request and corresponding information	Not asserted.

Popp et al. U.S. Patent No. 6,249,291 (195)

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
	transmitted from HTTP Server 206 to application 214. Application 214 can execute on the same or different server as CGIMessenger 210 and/or HTTP Server 206, for example. Application 214 executes an interaction flow to satisfy the user request. Application 214 can access an external data source such as database 224. Database 224 can be resident on the same server as application 214. Alternatively, database 224 can be resident on a separate server (e.g., a separate database server)." 7:24-35	
29(c): maintaining dynamic information regarding data sources a given page server may access	"At step 502, the request is received by the Web Objects server. Using the information contained in the request, the Web Objects server identifies the application that will process the request. At step 504, the Web Objects server determines whether the application is executing either on the same or a different server, for example. If the application is not currently loaded, the Web Objects server initiates the application at step 508 and processing continues at step 506. If the application is executing, processing continues to step 506." 27:45-54	Not asserted.
29(d)(1): dispatching said request to an appropriate page server based on said request and based on said dynamic information	See '554 1b5. " For more sophisticated requests, the present invention can execute logic such as CGIMessenger 210 and application 214, for example, to process a client request. The application preferably includes an object class hierarchy that can be implemented in any object-oriented language such as objective C, SmallTalk or JAVA. Objects (e.g., Web page definitional objects and control objects) dynamically generate a Web page and manage client interaction via the Internet and WWW." 7:50-58.	Not asserted.
29(d)(2): said page server receiving said request and	The request must be received to be processed	Not asserted.

Popp et al. U.S. Patent No. 6,249,291 (195)

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
29(d)(3): releasing said HTTP-compliant device to process other requests	TCP releasing	Not asserted.
29(e): processing said request, said processing being performed by said page server while said HTTP-compliant device concurrently processes said other requests, and	The web server and page server are different machines, so concurrent processing occurs inherently. "If an application is specified, CGIMessenger 210 transmits the client request and corresponding information transmitted from HTTP Server 206 to application 214. Application 214 can execute on the same or different server as CGIMessenger 210 and/or HTTP Server 206, for example. Application 214 executes an interaction flow to satisfy the user request. Application 214 can access an external data source such as database 224. Database 224 can be resident on the same server as application 214. Alternatively, database 224 can be resident on a separate server (e.g., a separate database server)." 7:25-35.	Not asserted.
29(f)(1): dynamically generating a Web page in response to said request	The application 214 returns "a Web page (e.g., an HTML Web page) that is dynamically generated using complex queries (or other data retrieval mechanisms) to retrieve data and dynamically generate an HTML page using complex logic." 7:44-58.	Not asserted.
29(f)(2): said page including data dynamically retrieved from one or more data sources	The contents of the Web page can contain data retrieved from an external data source (e.g., such as database server 318). 8:44-46	Not asserted.

Exhibit 9

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
1(a): A computer-implemented method for managing a dynamic Web page generation request to a Web server, said computer-implemented method comprising the steps of:	"Using the Oracle Web Listener, your web site can respond to client requests by generating HTML documents dynamically." p. 1-7. The "database is used for Web pages that are generated at runtime using "live" data." p. 1-2	Dr. Shamos did not prove that References 604 (Ex. 47) and 605 (Ex. 48) meet the preamble. In addition to the reasons set forth in this chart, my Response to Dr. Shamos's section on Oracle's Developments sets forth further reasons why References 604/605 do not anticipate any claims of the patents-in-suit.
1(b)(1): routing said request from said Web server to a page server	The Web server is Oracle's Web listener. The page server is a "WRB Executable Engine." "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), a program to be accessed through the CGI interface, or whether access to the file system of the machine on which the Listener resides is sufficient. If WRB access is required, the Listener passes the request to WRB Dispatcher for processing." p. 1-2. The WRB sends the request to an appropriate page server. Hence the required routing is performed	Under Oracle's proposed construction, the "page server" is "page generating software, running on a processor separate from that of the Web server, that generates dynamic Web pages." Dr. Shamos failed to prove that Reference 604/605 discloses a page server under Oracle's proposed construction because he failed to identify a "page server" that executes on a processor separate from the processor of the Web Server.
1(b)(2): said page server receiving said request and	The request must be received to be processed	
1(b)(3): releasing said Web server to process other requests	"AII inter-process communication is handled by the transport independent WRB protocol. Webserver 2.0 supports standard IPC mechanisms" Technical Note, p. 5. A standard IPC mechanism is Socket-Bind-Listen, hence SBL releasing is performed according to EpicRealm's contention	Dr. Shamos contended that the "SBL Theory" of Releasing disclosed by References 604/605 meets this limitation under either party's construction. I agree. Furthermore, the fact that Web Server 2.0 supports a standard IPC mechanism, does not necessarily mean that socket-bind-listen is used for communications.
1(b)(4): wherein said routing step further includes the steps of intercepting said request at	"The HTTP engine does not make any attempt at interpreting the URL, instead it is immediately handed off to the Web Request Broker for further	References 604 and 605 do not disclose "intercepting" for Oracle 2.0. As the quote cited by Dr. Shamos demonstrates, there is no decision by the "Web server"

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
said Web server	processing." Technical Note, p. 5. This "handing off" without processing is intercepting.	whether it should process a request or whether a "page server" should process a request. When "no suitable WRB Service is found, the request is passed back to the Web Listener and standard processing continues." (Ex. 48, Technical Note, p. 5). These References teach away from "intercepting" in the claimed invention. Specifically, the Oracle WebServer 2.0 Technical Note states: "[t]he typical HTTP API enables a custom extension to be linked into the HTTP server process and allows it to intercept and fulfill a client request." (Ex. 48, Technical Note, p. 3) However, Oracle 2.0 does not implement "the typical HTTP API" which allows intercepting because "the problem with the HTTP API approach is that all extensions are linked into the same executable and thus share the same memory, process, and signal handling space. … Oracle's answer to these issues is a new and improved architecture for the server, which effectively layers Oracle Web Server 2.0 into two fundamental components: a HTTP engine (the Oracle Web Listener) and a high-speed dispatch mechanism (the Web Request Broker) which routes requests to server extensions running in separate processes." Thus, not only do these References not disclose "intercepting" for Oracle 2.0, they teach away from it. Furthermore, Dr. Shamos failed to analyze these References under either epicRealm's or Oracle's proposed constructions.
1(b)(5): routing said request from said Web server to a dispatcher, and	The dispatcher is the "WRB Dispatcher." "Whenever the Web Listener receives a URL that calls for the WRB. it passes execution of the request to the WRB	As discussed in the section below, Dr. Shamos failed to establish that References 604 and 605 have a "dispatcher" because References 604 and 605 do not

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
	Dispatcher or simply Dispatcher. The Dispatcher maintains communication with a pool of processes called WRB Executable Engines (WRBXs).The Dispatcher finds a free WRBX that is configured to run the desired WRB Service and passes execution to it." p. 1-11	disclose "dispatching" under either Oracle's or epicRealm's proposed construction.
1(b)(6): dispatching said request to said page server	The WRB Dispatcher makes an informed selection of which page server (WRBX) should process the request based on dynamic information: which WRBX is free and which one is configured to run the desired service. Load balancing is also performed: "WRB Executable Engine (WRBX). One of a pool of processes that the WRB maintains continuously, so that HTTP requests requiring the execution of programs are not slowed down by the performance cost of spawning a new process. WRBX's are associated with WRB Cartridges and are created and destroyed according to workload." p. A-13. "The WRB dispatcher must decide what type of object is being requested. To do this, it examines the WRB configuration file, which maps virtual directories to WRB Services. ... The WRB Dispatcher performs dynamic load balancing between multiple instances of a WRB Service, and the webmaster can configure each WRB Service to have a minimum and maximum number of instances. An instance corresponds to a process in a UNIX environment." Technical Note, p. 5	References 604 and 605 do not disclose "dispatching" under epicRealm's or Oracle's construction. EpicRealm's proposed construction for "dispatching" is "examining a request to make an informed selection of which page server should process the request based on dynamic information maintained about page servers, the dynamic information indicating which page server can more efficiently process the request, and sending the request to the selected page server." References 604/605 do not disclose maintaining dynamic information indicating which page server can more efficiently process the request nor do References 604/605 disclose making an informed selection of which page server should process the request. The information used by the WRB dispatcher only indicates which WRBXs are executing a request and which are free to process other requests. A busy WRBX (one already processing a request) cannot process another request. The dispatcher does not maintain information about which one of the free WRBXs can more efficiently process a request. Thus, these References do not disclose maintaining dynamic information indicating which page server can more efficiently process the request.

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
		Dr. Shamos argued that the maintained dynamic information is "which WRBX is free and which one is configured to run the desired service." As discussed above, information about which WRBX is free is not information about which page server can more efficiently process a request. Further, information about which WRBX is configured to run a desired service is not dynamic information because the configuration information is stored in a static configuration file. (See also my response to Dr. Shamos's "Configuration File Dispatching.") Similarly, the information about which WRBX is configured to run a desired service is not information about which page server can more efficiently process a request because it does not differentiate between the relative efficiencies of page servers. A WRBX that is not configured to run a desired service is not an available page server to process a request, thus it is not considered in determining which page server can more efficiently process a request. In addition, these References do not disclose making an informed selection of which page server should process a request based on dynamic information. It is my understanding that the WRB dispatcher sends a request to a free WRBX that can process the request. This decision is not "informed" because it does not distinguish between the free page servers that can process a request. Similarly, Dr. Shamos also failed to prove that these References disclose "dispatching" under Oracle's construction. At a minimum, Dr. Shamos did not distinguish his analysis under epicRealm's and

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
		Oracle's proposed constructions for "dispatching." Therefore, for the reasons set forth above, these References do not disclose "dispatching." In addition, Oracle's construction requires that the "dispatcher" make an "informed selection of which page server should process the request…." In these References, there is not an "informed selection" for the reasons set forth above. Thus, these References do not disclose "dispatching" under Oracle's proposed construction. Dr. Shamos contended that these references disclose "load balancing" but failed to explain what relevance his "load balancing" has to the claim terms. Indeed, there are many forms of load balancing that do not meet the limitations of the claims.
1(c): processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests, and	Concurrency is taught expressly: "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), a program to be accessed through the CGI interface, or whether access to the file system of the machine on which the Listener resides is sufficient. If WRB access is required, the Listener passes the request to WRB Dispatcher for processing: then it returns to the task of listening for more incoming HTTP requests." p. 1-2. "Oracle's HTTP Server is designed to be a very scalable, high-performance network daemon. It is implemented as a multi-threaded, single-process asynchronous engine, allowing multiple concurrent requests to be processed at the same time using standard HTTP or HTTP over SSL" Technical Note, p. 3	

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
1(d)(1): dynamically generating a Web page in response to said request	"As documented in the CGI 1.1 specification, the Oracle Web Listener takes the contents of standard output and returns it to the Web browser that requested the dynamic HTML document." p. 6-6	
1(d)(2): said Web page including data dynamically retrieved from one or more data sources.	"The Oracle WebServer ... draws on information from the database and the operating system's (0S) file system as necessary to respond to the request. The file system can be used for static (hardcoded) Web pages, or form scripts that do not access the database, and the database is used for Web pages that are generated at runtime using "live" data." p. 1-2	
2: The computer-implemented method in claim 1 wherein said step of processing said request includes the step of identifying said one or more data sources from which to retrieve said data.	See '554 1b6 (configuration file)	References 604 and 605 cannot anticipate this claim because they do not anticipate the claim from which this claim depends.
3: The computer-implemented method in claim 2 wherein said step of dynamically generating said Web page includes the step of dynamically retrieving said data from said one or more data sources.	"PL/SQL cartridges. These execute stored PL/SQL procedures to generate HTML dynamically using Oracle data." p. 1-3. "If desired, you can use LiveHTML to run the PL/SOL Agent under CGI. and thereby incorporate dynamic Oracle data in hardcoded Web pages." p. 1-17	References 604 and 605 cannot anticipate this claim because they do not anticipate the claim from which this claim depends.
4: The computer-implemented method in claim 3 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or	"To do this, the Web Listener passes requests intended for these programs to the WRB Dispatcher, which maintains a pool of processes to which it can assign the requests." p. 1-7. This "pool of processes" is a connection cache	References 604 and 605 cannot anticipate this claim because they do not anticipate the claim from which this claim depends. Dr. Shamos failed to demonstrate that References 604 and 605 disclose "maintaining a connection cache to

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
more data sources.		said one or more data sources." Indeed, Dr. Shamos failed to identify the one or more data sources required by this claim. Further, a pool of processes is not a "connection cache" as one skilled in the art would understand that term.
5: The computer-implemented method in claim 3 wherein said step of processing said request includes the step of logging into said one or more data sources.	"A URL that invokes the PL/SQL Agent must specify a DCD (Database Connection Descriptor). This is an OS file maintained by the WebServer that provides the username, password, database, and other information to be used to establish the database connection." p. 1-14. "At the lowest level, Oracle; internal API allows us t o establish a connection to an Oracle7 Server using SQL*Net in one step, and actually logging on to the database in a separate step." Technical Note, p. 11	References 604 and 605 cannot anticipate this claim because they do not anticipate the claim from which this claim depends.
6: The computer-implemented method in claim 3 wherein said step of dynamically generating said Web page includes the step of maintaining a page cache containing said Web page.	"Page caching. Ordinarily, when the Web Listener opens a requested file, the file remains open and mapped into memory until all clients using the file are finished with it, at which point the Web Listener closes the file and frees the memory mapping associated with it. The Web Listener allows you to specify files to be cached. Cached files are opened when a client requests them, and remain open for the life of the Web Listener process. This optimizes access times for files, such as your home page, that are requested often." p. 1-5	Not asserted.
7: The computer-implemented method in claim 3 wherein said page server includes custom HTML extension templates for configuring said Web page.	"LiveHTML. A way to embed dynamic content in Web pages. This content can be either other Web pages or the output of scripts run by the Operating System. LiveHTML is an Oracle extension of the NCSA standard Server Side Includes functionality."	References 604 and 605 cannot anticipate this claim because they do not anticipate the claim from which this claim depends.

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
	p. 1-2. "The LiveHTML Interpreter enables you to include dynamic content in otherwise static Web pages" p. 1-16. "LiveHTML code is formatted as SGML comments ... The format for LiveHTML codes, therefore, is the following: < ! -- #command tag1="value1" tag2="value2" -- >. The tags are arguments to the commands, most of which actually only accept one of the possible tags." p. 4-30. "LiveHTML files supplement HTML with instructions that WebServer executes before transmitting the page. These instructions specify material that is to be included in the generated page. Said material can include other Web pages, environment variables, and the output of programs executed on the Server." p. A-8	
8: The computer-implemented method in claim 7 wherein said step of processing said request further includes the step of inserting said dynamically retrieved data from said one or more data sources into said custom HTML extension templates.	"include. This command specifies that a file is to be included in the generated HTML page at this point. The file can any of the following: - another LiveHTML file like the current one. - a regular HTML file. - an ASCII file." p. 4-30	References 604 and 605 cannot anticipate this claim because they do not anticipate the claim from which this claim depends.
9(a): A networked system for managing a dynamic Web page generation request, said system comprising	OWS 2.0 operates with a Web server, hence is networked	Dr. Shamos did not prove that References 604 and 605 meet the preamble.
9(b): one or more data sources	"The Oracle WebServer ... draws on information from the database and the operating system's (0S) file system as necessary to respond to the request. The file system can be used for static (hardcoded) Web	

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
	pages, or form scripts that do not access the database, and the database is used for Web pages that are generated at runtime using "live" data." p. 1-2	
9(c): a page server having a processing means	The Web server is Oracle's Web listener. The page server is a "WRB Executable Engine." "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), a program to be accessed through the CGI interface, or whether access to the file system of the machine on which the Listener resides is sufficient. If WRB access is required, the Listener passes the request to WRB Dispatcher for processing." p. 1-2. The WRB sends the request to an appropriate page server. Hence the required routing is performed	Under Oracle's proposed construction, the "page server" is "page generating software, running on a processor separate from that of the second computer system, that generates dynamic Web pages." Dr. Shamos failed to prove that Reference 604/605 discloses a page server under Oracle's proposed construction because he failed to identify a "page server" that executes on a processor separate from the processor of the second computer system. Dr. Shamos failed to identify the specified function and the corresponding structure, material or acts for the "processing means" as required by 35 U.S.C. § 112 ¶ 6. Furthermore, Dr. Shamos failed to demonstrate that Reference 604/605 discloses the specified function and corresponding structure, material or acts for the "processing means" or equivalents thereof.
9(d): a first computer system including means for generating said request, and	The first computer system includes the user's web browser. "From a web browser's point of view, Oracle WebServer 2.0 looks just like any other web server." Technical Note, p. 2	Dr. Shamos failed to identify the specified function and the corresponding structure, material or acts for the "means for generating said request" as required by 35 U.S.C. § 112 ¶ 6. Furthermore, Dr. Shamos failed to demonstrate that Reference 604/605 discloses the specified function and corresponding structure, material or acts for the "means for generating said request" or equivalents thereof.
9(e)(1): a second computer system including	The second computer system includes the Web listener, the Web Request Broker, as shown on Technical Note p. 2, and the WRB Executable Engines (page servers)	

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
9(e)(2): means for receiving said request from said first computer	The Web server is Oracle's Web listener. The page server is a "WRB Executable Engine." "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), a program to be accessed through the CGI interface, or whether access to the file system of the machine on which the Listener resides is sufficient. If WRB access is required, the Listener passes the request to WRB Dispatcher for processing." p. 1-2. The WRB sends the request to an appropriate page server. Hence the required routing is performed	Dr. Shamos failed to identify the specified function and the corresponding structure, material or acts for the "means for receiving" as required by 35 U.S.C. § 112 ¶ 6. Furthermore, Dr. Shamos failed to demonstrate that Reference 604/605 discloses the specified function and corresponding structure, material or acts for the "means for receiving" or equivalents thereof.
9(e)(3): said second computer system also including a router, said router routing said request from said second computer system to said page server	The Web server is Oracle's Web listener. The page server is a "WRB Executable Engine." "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), a program to be accessed through the CGI interface, or whether access to the file system of the machine on which the Listener resides is sufficient. If WRB access is required, the Listener passes the request to WRB Dispatcher for processing." p. 1-2. The WRB sends the request to an appropriate page server. Hence the required routing is performed	Under Oracle's proposed construction, the "page server" is "page generating software, running on a processor separate from that of the Web server, that generates dynamic Web pages." Dr. Shamos failed to prove that Reference 604/605 discloses a page server under Oracle's proposed construction because he failed to identify a "page server" that executes on a processor separate from the processor of the Web Server.
9(e)(4): wherein said routing further includes intercepting said request at said second computer	"The HTTP engine does not make any attempt at interpreting the URL, instead it is immediately handed off to the Web Request Broker for further processing." Technical Note, p. 5. This "handing off" without processing is intercepting.	References 604 and 605 do not disclose "intercepting" for Oracle 2.0. As the quote cited by Dr. Shamos demonstrates, there is no decision by the "Web server" whether it should process a request or whether a "page server" should process a request. When "no suitable WRB Service is found, the request is passed back to

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
		the Web Listener and standard processing continues." (Ex. 48, Technical Note, p. 5). These References teach away from "intercepting" in the claimed invention. Specifically, the Oracle WebServer 2.0 Technical Note states: "[t]he typical HTTP API enables a custom extension to be linked into the HTTP server process and allows it to intercept and fulfill a client request." (Ex. 48, Technical Note, p. 3) However, Oracle 2.0 does not implement "the typical HTTP API" which allows intercepting because "the problem with the HTTP API approach is that all extensions are linked into the same executable and thus share the same memory, process, and signal handling space. … Oracle's answer to these issues is a new and improved architecture for the server, which effectively layers Oracle Web Server 2.0 into two fundamental components: a HTTP engine (the Oracle Web Listener) and a high-speed dispatch mechanism (the Web Request Broker) which routes requests to server extensions running in separate processes." Thus, not only do these References not disclose "intercepting" for Oracle 2.0, they teach away from it. Furthermore, Dr. Shamos failed to analyze these References under either epicRealm's or Oracle's proposed constructions.
9(e)(5): routing said request from said second computer to a dispatcher, and	The dispatcher is the "WRB Dispatcher." "Whenever the Web Listener receives a URL that calls for the WRB. it passes execution of the request to the WRB Dispatcher or simply Dispatcher. The Dispatcher maintains communication with a pool of processes called WRB Executable Engines (WRBXs).The	As discussed in the section below, Dr. Shamos failed to establish that References 604 and 605 have a "dispatcher" because References 604 and 605 do not disclose "dispatching" under either Oracle's or epicRealm's proposed construction.

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
	Dispatcher finds a free WRBX that is configured to run the desired WRB Service and passes execution to it." p. 1-11	
9(e)(6): dispatching said request to said page server	The WRB Dispatcher makes an informed selection of which page server (WRBX) should process the request based on dynamic information: which WRBX is free and which one is configured to run the desired service. Load balancing is also performed: "WRB Executable Engine (WRBX). One of a pool of processes that the WRB maintains continuously, so that HTTP requests requiring the execution of programs are not slowed down by the performance cost of spawning a new process. WRBX's are associated with WRB Cartridges and are created and destroyed according to workload." p. A-13. "The WRB dispatcher must decide what type of object is being requested. To do this, it examines the WRB configuration file, which maps virtual directories to WRB Services. ... The WRB Dispatcher performs dynamic load balancing between multiple instances of a WRB Service, and the webmaster can configure each WRB Service to have a minimum and maximum number of instances. An instance corresponds to a process in a UNIX environment." Technical Note, p. 5	References 604 and 605 do not disclose "dispatching" under epicRealm's or Oracle's construction. EpicRealm's proposed construction for "dispatching" is "examining a request to make an informed selection of which page server should process the request based on dynamic information maintained about page servers, the dynamic information indicating which page server can more efficiently process the request, and sending the request to the selected page server." References 604/605 do not disclose maintaining dynamic information indicating which page server can more efficiently process the request nor do References 604/605 disclose making an informed selection of which page server should process the request. The information used by the WRB dispatcher only indicates which WRBXs are executing a request and which are free to process other requests. A busy WRBX (one already processing a request) cannot process another request. The dispatcher does not maintain information about which one of the free WRBXs can more efficiently process a request. Thus, these References do not disclose maintaining dynamic information indicating which page server can more efficiently process the request. Dr. Shamos argued that the maintained dynamic information is "which WRBX is free and which one is configured to run the desired service." As discussed

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
		above, information about which WRBX is free is not information about which page server can more efficiently process a request. Further, information about which WRBX is configured to run a desired service is not dynamic information because the configuration information is stored in a static configuration file. (See also my response to Dr. Shamos's "Configuration File Dispatching.") Similarly, the information about which WRBX is configured to run a desired service is not information about which page server can more efficiently process a request because it does not differentiate between the relative efficiencies of page servers. A WRBX that is not configured to run a desired service is not an available page server to process a request, thus it is not considered in determining which page server can more efficiently process a request. In addition, these References do not disclose making an informed selection of which page server should process a request based on dynamic information. It is my understanding that the WRB dispatcher sends a request to a free WRBX that can process the request. This decision is not "informed" because it does not distinguish between the free page servers that can process a request. Similarly, Dr. Shamos also failed to prove that these References disclose "dispatching" under Oracle's construction. At a minimum, Dr. Shamos did not distinguish his analysis under epicRealm's and Oracle's proposed constructions for "dispatching." Therefore, for the reasons set forth above, these References do not disclose "dispatching." In addition,

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
		Oracle's construction requires that the "dispatcher" make an "informed selection of which page server should process the request…." In these References, there is not an "informed selection" for the reasons set forth above. Thus, these References do not disclose "dispatching" under Oracle's proposed construction. Dr. Shamos contended that these references disclose "load balancing" but failed to explain what relevance his "load balancing" has to the claim terms. Indeed, there are many forms of load balancing that do not meet the limitations of the claims.
9(e)(7): said page server receiving said request and	The request must be received to be processed	
9(e)(8): releasing said second computer system to process other requests	"AII inter-process communication is handled by the transport independent WRB protocol. Webserver 2.0 supports standard IPC mechanisms" Technical Note, p. 5. A standard IPC mechanism is Socket-Bind-Listen, hence SBL releasing is performed according to EpicRealm's contention	Dr. Shamos contended that the "SBL Theory" of Releasing disclosed by References 604/605 meets this limitation under either party's construction. I agree. Furthermore, the fact that Web Server 2.0 supports a standard IPC mechanism, does not necessarily mean that socket-bind-listen is used for communications.
9(e)(9): said page server processing means processing said request and dynamically generating a Web page in response to said request	"As documented in the CGI 1.1 specification, the Oracle Web Listener takes the contents of standard output and returns it to the Web browser that requested the dynamic HTML document." p. 6-6	
9(e)(10): said Web page including data dynamically retrieved from said one or more data sources	"The Oracle WebServer ... draws on information from the database and the operating system's (0S) file system as necessary to respond to the request. The file system can be used for static (hardcoded) Web pages, or form scripts that do not access the database, and the database is used for Web pages that are generated at runtime using "live" data." p. 1-2	

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
10(a): The networked system in claim 9 wherein said router in said second computer system includes:	OWS 2.0 operates with a Web server, hence is networked	References 604 and 605 cannot anticipate this claim because they do not anticipate the claim from which this claim depends.
10(b): an interceptor intercepting said request at said second computer system and routing said request, and	"The HTTP engine does not make any attempt at interpreting the URL, instead it is immediately handed off to the Web Request Broker for further processing." Technical Note, p. 5. This "handing off" without processing is intercepting.	References 604 and 605 cannot anticipate this claim because they do not anticipate the claim from which this claim depends. References 604 and 605 do not disclose "intercepting" for Oracle 2.0. As the quote cited by Dr. Shamos demonstrates, there is no decision by the "Web server" whether it should process a request or whether a "page server" should process a request. When "no suitable WRB Service is found, the request is passed back to the Web Listener and standard processing continues." (Ex. 48, Technical Note, p. 5). These References teach away from "intercepting" in the claimed invention. Specifically, the Oracle WebServer 2.0 Technical Note states: "[t]he typical HTTP API enables a custom extension to be linked into the HTTP server process and allows it to intercept and fulfill a client request." (Ex. 48, Technical Note, p. 3) However, Oracle 2.0 does not implement "the typical HTTP API" which allows intercepting because "the problem with the HTTP API approach is that all extensions are linked into the same executable and thus share the same memory, process, and signal handling space. … Oracle's answer to these issues is a new and improved architecture for the server, which effectively layers Oracle Web Server 2.0 into two fundamental components: a HTTP engine (the Oracle Web Listener)

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
		and a high-speed dispatch mechanism (the Web Request Broker) which routes requests to server extensions running in separate processes." Thus, not only do these References not disclose "intercepting" for Oracle 2.0, they teach away from it. Furthermore, Dr. Shamos failed to analyze these References under either epicRealm's or Oracle's proposed constructions.
10(c): a dispatcher receiving said routed request from said interceptor and dispatching said request to said page server	The WRB Dispatcher makes an informed selection of which page server (WRBX) should process the request based on dynamic information: which WRBX is free and which one is configured to run the desired service. Load balancing is also performed: "WRB Executable Engine (WRBX). One of a pool of processes that the WRB maintains continuously, so that HTTP requests requiring the execution of programs are not slowed down by the performance cost of spawning a new process. WRBX's are associated with WRB Cartridges and are created and destroyed according to workload." p. A-13. "The WRB dispatcher must decide what type of object is being requested. To do this, it examines the WRB configuration file, which maps virtual directories to WRB Services. ... The WRB Dispatcher performs dynamic load balancing between multiple instances of a WRB Service, and the webmaster can configure each WRB Service to have a minimum and maximum number of instances. An instance corresponds to a process in a UNIX environment." Technical Note, p. 5	References 604 and 605 cannot anticipate this claim because they do not anticipate the claim from which this claim depends. References 604 and 605 do not disclose "dispatching" under epicRealm's or Oracle's construction. References 604 and 605 do not disclose "dispatching" under epicRealm's or Oracle's construction. EpicRealm's proposed construction for "dispatching" is "examining a request to make an informed selection of which page server should process the request based on dynamic information maintained about page servers, the dynamic information indicating which page server can more efficiently process the request, and sending the request to the selected page server." References 604/605 do not disclose maintaining dynamic information indicating which page server can more efficiently process the request nor do References 604/605 disclose making an informed selection of which page server should process the request. The information used by the WRB dispatcher only indicates which WRBXs are executing a request and which are

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
		free to process other requests. A busy WRBX (one already processing a request) cannot process another request. The dispatcher does not maintain information about which one of the free WRBXs can more efficiently process a request. Thus, these References do not disclose maintaining dynamic information indicating which page server can more efficiently process the request. Dr. Shamos argued that the maintained dynamic information is "which WRBX is free and which one is configured to run the desired service." As discussed above, information about which WRBX is free is not information about which page server can more efficiently process a request. Further, information about which WRBX is configured to run a desired service is not dynamic information because the configuration information is stored in a static configuration file. (See also my response to Dr. Shamos's "Configuration File Dispatching.") Similarly, the information about which WRBX is configured to run a desired service is not information about which page server can more efficiently process a request because it does not differentiate between the relative efficiencies of page servers. A WRBX that is not configured to run a desired service is not an available page server to process a request, thus it is not considered in determining which page server can more efficiently process a request. In addition, these References do not disclose making an informed selection of which page server should process a request based on dynamic information. It is my understanding that the WRB dispatcher sends a request

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
		to a free WRBX that can process the request. This decision is not "informed" because it does not distinguish between the free page servers that can process a request. Similarly, Dr. Shamos also failed to prove that these References disclose "dispatching" under Oracle's construction. At a minimum, Dr. Shamos did not distinguish his analysis under epicRealm's and Oracle's proposed constructions for "dispatching." Therefore, for the reasons set forth above, these References do not disclose "dispatching." In addition, Oracle's construction requires that the "dispatcher" make an "informed selection of which page server should process the request…." In these References, there is not an "informed selection" for the reasons set forth above. Thus, these References do not disclose "dispatching" under Oracle's proposed construction. Dr. Shamos contended that these references disclose "load balancing" but failed to explain what relevance his "load balancing" has to the claim terms. Indeed, there are many forms of load balancing that do not meet the limitations of the claims.
11(a): A machine readable medium having stored thereon data representing sequences of instructions, which when executed by a computer system, cause said computer system to perform the steps of:	OWS 2.0is a programmed system whose instructions are on a machine readable medium	Dr. Shamos did not prove that References 604 and 605 meet the preamble.
11(b)(1): routing a dynamic Web page generation request	The Web server is Oracle's Web listener. The page server is a "WRB Executable Engine." "When the	Under Oracle's proposed construction, the "page server" is "page generating software, running on a

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
from a Web server to a page server	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), a program to be accessed through the CGI interface, or whether access to the file system of the machine on which the Listener resides is sufficient. If WRB access is required, the Listener passes the request to WRB Dispatcher for processing." p. 1-2. The WRB sends the request to an appropriate page server. Hence the required routing is performed	processor separate from that of the Web server, that generates dynamic Web pages." Dr. Shamos failed to prove that Reference 604/605 discloses a page server under Oracle's proposed construction because he failed to identify a "page server" that executes on a processor separate from the processor of the Web Server.
11(b)(2): said page server receiving said request and	The request must be received to be processed	
11(b)(3): releasing said Web server to process other requests	"AII inter-process communication is handled by the transport independent WRB protocol. Webserver 2.0 supports standard IPC mechanisms" Technical Note, p. 5. A standard IPC mechanism is Socket-Bind-Listen, hence SBL releasing is performed according to EpicRealm's contention	Dr. Shamos contended that the "SBL Theory" of Releasing disclosed by References 604/605 meets this limitation under either party's construction. I agree. Furthermore, the fact that Web Server 2.0 supports a standard IPC mechanism, does not necessarily mean that socket-bind-listen is used for communications.
11(b)(4): wherein said routing step further includes the steps of intercepting said request at said Web server	"The HTTP engine does not make any attempt at interpreting the URL, instead it is immediately handed off to the Web Request Broker for further processing." Technical Note, p. 5. This "handing off" without processing is intercepting.	References 604 and 605 do not disclose "intercepting" for Oracle 2.0. As the quote cited by Dr. Shamos demonstrates, there is no decision by the "Web server" whether it should process a request or whether a "page server" should process a request. When "no suitable WRB Service is found, the request is passed back to the Web Listener and standard processing continues." (Ex. 48, Technical Note, p. 5). These References teach away from "intercepting" in the claimed invention. Specifically, the Oracle WebServer 2.0 Technical Note states: "[t]he typical HTTP API enables a custom extension to be linked into

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
		the HTTP server process and allows it to intercept and fulfill a client request." (Ex. 48, Technical Note, p. 3) However, Oracle 2.0 does not implement "the typical HTTP API" which allows intercepting because "the problem with the HTTP API approach is that all extensions are linked into the same executable and thus share the same memory, process, and signal handling space. … Oracle's answer to these issues is a new and improved architecture for the server, which effectively layers Oracle Web Server 2.0 into two fundamental components: a HTTP engine (the Oracle Web Listener) and a high-speed dispatch mechanism (the Web Request Broker) which routes requests to server extensions running in separate processes." Thus, not only do these References not disclose "intercepting" for Oracle 2.0, they teach away from it. Furthermore, Dr. Shamos failed to analyze these References under either epicRealm's or Oracle's proposed constructions.
11(b)(5): routing said request from said Web server to a dispatcher, and	The dispatcher is the "WRB Dispatcher." "Whenever the Web Listener receives a URL that calls for the WRB. it passes execution of the request to the WRB Dispatcher or simply Dispatcher. The Dispatcher maintains communication with a pool of processes called WRB Executable Engines (WRBXs).The Dispatcher finds a free WRBX that is configured to run the desired WRB Service and passes execution to it." p. 1-11	As discussed in the section below, Dr. Shamos failed to establish that References 604 and 605 have a "dispatcher" because References 604 and 605 do not disclose "dispatching" under either Oracle's or epicRealm's proposed construction.
11(b)(6): dispatching said request to said page server	The WRB Dispatcher makes an informed selection of which page server (WRBX) should process the request based on dynamic information: which WRBX	References 604 and 605 do not disclose "dispatching" under epicRealm's or Oracle's construction.

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
	is free and which one is configured to run the desired service. Load balancing is also performed: "WRB Executable Engine (WRBX). One of a pool of processes that the WRB maintains continuously, so that HTTP requests requiring the execution of programs are not slowed down by the performance cost of spawning a new process. WRBX's are associated with WRB Cartridges and are created and destroyed according to workload." p. A-13. "The WRB dispatcher must decide what type of object is being requested. To do this, it examines the WRB configuration file, which maps virtual directories to WRB Services. ... The WRB Dispatcher performs dynamic load balancing between multiple instances of a WRB Service, and the webmaster can configure each WRB Service to have a minimum and maximum number of instances. An instance corresponds to a process in a UNIX environment." Technical Note, p. 5	EpicRealm's proposed construction for "dispatching" is "examining a request to make an informed selection of which page server should process the request based on dynamic information maintained about page servers, the dynamic information indicating which page server can more efficiently process the request, and sending the request to the selected page server." References 604/605 do not disclose maintaining dynamic information indicating which page server can more efficiently process the request nor do References 604/605 disclose making an informed selection of which page server should process the request. The information used by the WRB dispatcher only indicates which WRBXs are executing a request and which are free to process other requests. A busy WRBX (one already processing a request) cannot process another request. The dispatcher does not maintain information about which one of the free WRBXs can more efficiently process a request. Thus, these References do not disclose maintaining dynamic information indicating which page server can more efficiently process the request. Dr. Shamos argued that the maintained dynamic information is "which WRBX is free and which one is configured to run the desired service." As discussed above, information about which WRBX is free is not information about which page server can more efficiently process a request. Further, information about which WRBX is configured to run a desired service is not dynamic information because the configuration information is stored in a static configuration file. (See also my response to Dr. Shamos's "Configuration File Dispatching.")

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
		Similarly, the information about which WRBX is configured to run a desired service is not information about which page server can more efficiently process a request because it does not differentiate between the relative efficiencies of page servers. A WRBX that is not configured to run a desired service is not an available page server to process a request, thus it is not considered in determining which page server can more efficiently process a request. In addition, these References do not disclose making an informed selection of which page server should process a request based on dynamic information. It is my understanding that the WRB dispatcher sends a request to a free WRBX that can process the request. This decision is not "informed" because it does not distinguish between the free page servers that can process a request. Similarly, Dr. Shamos also failed to prove that these References disclose "dispatching" under Oracle's construction. At a minimum, Dr. Shamos did not distinguish his analysis under epicRealm's and Oracle's proposed constructions for "dispatching." Therefore, for the reasons set forth above, these References do not disclose "dispatching." In addition, Oracle's construction requires that the "dispatcher" make an "informed selection of which page server should process the request…." In these References, there is not an "informed selection" for the reasons set forth above. Thus, these References do not disclose "dispatching" under Oracle's proposed construction. Dr. Shamos contended that these references disclose

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
		"load balancing" but failed to explain what relevance his "load balancing" has to the claim terms. Indeed, there are many forms of load balancing that do not meet the limitations of the claims.
11(c): processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests, and	Concurrency is taught expressly: "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), a program to be accessed through the CGI interface, or whether access to the file system of the machine on which the Listener resides is sufficient. If WRB access is required, the Listener passes the request to WRB Dispatcher for processing: then it returns to the task of listening for more incoming HTTP requests." p. 1-2. "Oracle's HTTP Server is designed to be a very scalable, high-performance network daemon. It is implemented as a multi-threaded, single-process asynchronous engine, allowing multiple concurrent requests to be processed at the same time using standard HTTP or HTTP over SSL" Technical Note, p. 3	
11(d): dynamically generating a Web page, said Web page including data retrieved from one or more data sources	"As documented in the CGI 1.1 specification, the Oracle Web Listener takes the contents of standard output and returns it to the Web browser that requested the dynamic HTML document." p. 6-6	Dr. Shamos only contended that References 604 and 605 disclose "dynamically generating a Web page in response to said request." Dr. Shamos failed to contend that References 604 and 605 disclose "dynamically generating a Web page, said Web page including data retrieved from one or more data sources."

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
2[1(a)]: A computer-implemented method for managing a dynamic Web page generation request to a Web server, said computer-implemented method comprising the steps of:	"Using the Oracle Web Listener, your web site can respond to client requests by generating HTML documents dynamically." p. 1-7. The "database is used for Web pages that are generated at runtime using "live" data." p. 1-2	Dr. Shamos did not prove that References 604 and 605 meet the preamble.
2[1(b)(1)]: routing a request from a Web server to a page server	The Web server is Oracle's Web listener. The page server is a "WRB Executable Engine." "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), a program to be accessed through the CGI interface, or whether access to the file system of the machine on which the Listener resides is sufficient. If WRB access is required, the Listener passes the request to WRB Dispatcher for processing." p. 1-2. The WRB sends the request to an appropriate page server. Hence the required routing is performed	Under Oracle's proposed construction, the "page server" is "page generating software, running on a processor separate from that of the Web server, that generates dynamic Web pages." Dr. Shamos failed to prove that Reference 604/605 discloses a page server under Oracle's proposed construction because he failed to identify a "page server" that executes on a processor separate from the processor of the Web Server.
2[1(b)(2)]: said page server receiving said request and	The request must be received to be processed	
2[1(b)(3)]: releasing said Web server to process other requests	"AII inter-process communication is handled by the transport independent WRB protocol. Webserver 2.0 supports standard IPC mechanisms" Technical Note, p. 5. A standard IPC mechanism is Socket-Bind-Listen, hence SBL releasing is performed according to EpicRealm's contention	Dr. Shamos contended that the "SBL Theory" of Releasing disclosed by References 604/605 meets this limitation under either party's construction. I agree. Furthermore, the fact that Web Server 2.0 supports a standard IPC mechanism, does not necessarily mean that socket-bind-listen is used for communications.
2[1(c)(1)]: wherein said routing step further includes the steps of: intercepting said	"The HTTP engine does not make any attempt at interpreting the URL, instead it is immediately handed off to the Web Request Broker for further	References 604 and 605 do not disclose "intercepting" for Oracle 2.0. As the quote cited by Dr. Shamos demonstrates, there is no decision by the "Web server"

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
request at said Web server and	processing." Technical Note, p. 5. This "handing off" without processing is intercepting.	whether it should process a request or whether a "page server" should process a request. When "no suitable WRB Service is found, the request is passed back to the Web Listener and standard processing continues." (Ex. 48, Technical Note, p. 5). These References teach away from "intercepting" in the claimed invention. Specifically, the Oracle WebServer 2.0 Technical Note states: "[t]he typical HTTP API enables a custom extension to be linked into the HTTP server process and allows it to intercept and fulfill a client request." (Ex. 48, Technical Note, p. 3) However, Oracle 2.0 does not implement "the typical HTTP API" which allows intercepting because "the problem with the HTTP API approach is that all extensions are linked into the same executable and thus share the same memory, process, and signal handling space. … Oracle's answer to these issues is a new and improved architecture for the server, which effectively layers Oracle Web Server 2.0 into two fundamental components: a HTTP engine (the Oracle Web Listener) and a high-speed dispatch mechanism (the Web Request Broker) which routes requests to server extensions running in separate processes." Thus, not only do these References not disclose "intercepting" for Oracle 2.0, they teach away from it. Furthermore, Dr. Shamos failed to analyze these References under either epicRealm's or Oracle's proposed constructions.
2[1(c)(2)]: routing said request to said page server	The Web server is Oracle's Web listener. The page server is a "WRB Executable Engine." "When the Web Listener receives a URL, it determines whether	

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
	the request requires the use of a Service to be accessed through the Web Request Broker (WRB), a program to be accessed through the CGI interface, or whether access to the file system of the machine on which the Listener resides is sufficient. If WRB access is required, the Listener passes the request to WRB Dispatcher for processing." p. 1-2. The WRB sends the request to an appropriate page server. Hence the required routing is performed	
2[1(d)]: processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests, and	Concurrency is taught expressly: "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), a program to be accessed through the CGI interface, or whether access to the file system of the machine on which the Listener resides is sufficient. If WRB access is required, the Listener passes the request to WRB Dispatcher for processing: then it returns to the task of listening for more incoming HTTP requests." p. 1-2. "Oracle's HTTP Server is designed to be a very scalable, high-performance network daemon. It is implemented as a multi-threaded, single-process asynchronous engine, allowing multiple concurrent requests to be processed at the same time using standard HTTP or HTTP over SSL" Technical Note, p. 3	
2[1(e)(1)]: dynamically generating a Web page in response to said request	"As documented in the CGI 1.1 specification, the Oracle Web Listener takes the contents of standard output and returns it to the Web browser that requested the dynamic HTML document." p. 6-6	

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
2[1(e)(2)]: said Web page including data dynamically retrieved from one or more data sources	"The Oracle WebServer ... draws on information from the database and the operating system's (0S) file system as necessary to respond to the request. The file system can be used for static (hardcoded) Web pages, or form scripts that do not access the database, and the database is used for Web pages that are generated at runtime using "live" data." p. 1-2	
2(a): The computer-implemented method in claim 1 wherein said step of routing said request includes the steps of:	"Using the Oracle Web Listener, your web site can respond to client requests by generating HTML documents dynamically." p. 1-7. The "database is used for Web pages that are generated at runtime using "live" data." p. 1-2	Dr. Shamos did not prove that References 604 and 605 meet the preamble.
2(b): routing said request from said Web server to a dispatcher, and	The dispatcher is the "WRB Dispatcher." "Whenever the Web Listener receives a URL that calls for the WRB. it passes execution of the request to the WRB Dispatcher or simply Dispatcher. The Dispatcher maintains communication with a pool of processes called WRB Executable Engines (WRBXs).The Dispatcher finds a free WRBX that is configured to run the desired WRB Service and passes execution to it." p. 1-11	As discussed in the section below, Dr. Shamos failed to establish that References 604 and 605 have a "dispatcher" because References 604 and 605 do not disclose "dispatching" under either Oracle's or epicRealm's proposed construction.
2(c): dispatching said request to said page server	The WRB Dispatcher makes an informed selection of which page server (WRBX) should process the request based on dynamic information: which WRBX is free and which one is configured to run the desired service. Load balancing is also performed: "WRB Executable Engine (WRBX). One of a pool of processes that the WRB maintains continuously, so that HTTP requests requiring the execution of programs are not slowed down by the performance	References 604 and 605 do not disclose "dispatching" under epicRealm's or Oracle's construction. EpicRealm's proposed construction for "dispatching" is "examining a request to make an informed selection of which page server should process the request based on dynamic information maintained about page servers, the dynamic information indicating which page server can more efficiently process the request, and sending

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
	cost of spawning a new process. WRBX's are associated with WRB Cartridges and are created and destroyed according to workload." p. A-13. "The WRB dispatcher must decide what type of object is being requested. To do this, it examines the WRB configuration file, which maps virtual directories to WRB Services. ... The WRB Dispatcher performs dynamic load balancing between multiple instances of a WRB Service, and the webmaster can configure each WRB Service to have a minimum and maximum number of instances. An instance corresponds to a process in a UNIX environment." Technical Note, p. 5	the request to the selected page server." References 604/605 do not disclose maintaining dynamic information indicating which page server can more efficiently process the request nor do References 604/605 disclose making an informed selection of which page server should process the request. The information used by the WRB dispatcher only indicates which WRBXs are executing a request and which are free to process other requests. A busy WRBX (one already processing a request) cannot process another request. The dispatcher does not maintain information about which one of the free WRBXs can more efficiently process a request. Thus, these References do not disclose maintaining dynamic information indicating which page server can more efficiently process the request. Dr. Shamos argued that the maintained dynamic information is "which WRBX is free and which one is configured to run the desired service." As discussed above, information about which WRBX is free is not information about which page server can more efficiently process a request. Further, information about which WRBX is configured to run a desired service is not dynamic information because the configuration information is stored in a static configuration file. (See also my response to Dr. Shamos's "Configuration File Dispatching.") Similarly, the information about which WRBX is configured to run a desired service is not information about which page server can more efficiently process a request because it does not differentiate between the relative efficiencies of page servers. A WRBX that is not configured to run a desired service is not an

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
		available page server to process a request, thus it is not considered in determining which page server can more efficiently process a request. In addition, these References do not disclose making an informed selection of which page server should process a request based on dynamic information. It is my understanding that the WRB dispatcher sends a request to a free WRBX that can process the request. This decision is not "informed" because it does not distinguish between the free page servers that can process a request. Similarly, Dr. Shamos also failed to prove that these References disclose "dispatching" under Oracle's construction. At a minimum, Dr. Shamos did not distinguish his analysis under epicRealm's and Oracle's proposed constructions for "dispatching." Therefore, for the reasons set forth above, these References do not disclose "dispatching." In addition, Oracle's construction requires that the "dispatcher" make an "informed selection of which page server should process the request…." In these References, there is not an "informed selection" for the reasons set forth above. Thus, these References do not disclose "dispatching" under Oracle's proposed construction. Dr. Shamos contended that these references disclose "load balancing" but failed to explain what relevance his "load balancing" has to the claim terms. Indeed, there are many forms of load balancing that do not meet the limitations of the claims.
3: The computer-implemented	See '554 1b6 (configuration file)	Not asserted.

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
method in claim 1 wherein said step of processing said request includes the step of identifying said one or more data sources from which to retrieve said data		
4: The computer-implemented method in claim 1 wherein said step of dynamically generating said Web page includes the step of dynamically retrieving said data from said one or more data sources	"PL/SQL cartridges. These execute stored PL/SQL procedures to generate HTML dynamically using Oracle data." p. 1-3. "If desired, you can use LiveHTML to run the PL/SOL Agent under CGI. and thereby incorporate dynamic Oracle data in hardcoded Web pages." p. 1-17	Not asserted.
5: The computer-implemented method in claim 1 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more data sources	"To do this, the Web Listener passes requests intended for these programs to the WRB Dispatcher, which maintains a pool of processes to which it can assign the requests." p. 1-7. This "pool of processes" is a connection cache	Not asserted.
6: The computer-implemented method in claim 1 wherein said step of processing said request includes the step of logging into said one or more data sources	"A URL that invokes the PL/SQL Agent must specify a DCD (Database Connection Descriptor). This is an OS file maintained by the WebServer that provides the username, password, database, and other information to be used to establish the database connection." p. 1-14. "At the lowest level, Oracle; internal API allows us t o establish a connection to an Oracle7 Server using SQL*Net in one step, and actually logging on to the database in a separate step." Technical Note, p. 11	Not asserted.
7: The computer-implemented method in claim 1 wherein said	"Page caching. Ordinarily, when the Web Listener opens a requested file, the file remains open and	Not asserted.

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
step of dynamically generating said Web page includes the step of maintaining a page cache containing said Web page	mapped into memory until all clients using the file are finished with it, at which point the Web Listener closes the file and frees the memory mapping associated with it. The Web Listener allows you to specify files to be cached. Cached files are opened when a client requests them, and remain open for the life of the Web Listener process. This optimizes access times for files, such as your home page, that are requested often." p. 1-5	
8: The computer-implemented method in claim 1 wherein said page server includes tag-based text templates for configuring said Web page	LiveHTML comprises tag-based text templates. See '554 7	Not asserted.
9: The computer-implemented method in claim 8 wherein said step of processing said request further includes the step of inserting said-dynamically retrieved data from said one or more data sources into said tag-based text templates	See '554 8.	Not asserted.
10: The computer-implemented method in claim 8 wherein at least one of said tag-based text templates drives a format of the data dynamically retrieved from said one or more data sources in response to said request	Section 5 (e.g., page 5-5, MAC003147) describes the use of tag-based text templates in PL/SQL procedures to drive the format of the data in creating the generated page. "The Oracle WebServer Developer's Toolkit is a set of procedures that help the customer create programs to generate dynamic documents that are easily presented in HTML." p. 1-2 (MAC003052). HTML determines the format of a web page	Not asserted.

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
11: The computer-implemented method in claim 8 wherein said tag-based text templates include HTML templates	See '554 8.	Not asserted.
12: The computer-implemented method in claim 1 wherein said step of processing said request further includes the step of dynamically updating data at said one or more data sources	"You can also use a cursor to update a table's contents. To do this, use a FOR UPDATE clause to lock the rows in the table. See "Using FOR UPDATE" in the PL/SQL User's Guide and Reference for more information. Sometimes, you may need to use cursor variables. which are not associated with a query until runtime. This is a form of dynamic SQL." p. B-19	Not asserted.
13: The computer-implemented method in claim 1 wherein said step of processing said request further includes the step of processing an object handling extension	"The Java Interpreter includes classes for native Oracle7 access via PL/SQL and dynamic creation of HTML documents. For customers who wish to employ their own programming language of choice, Oracle WebServer 2.0 offers an easy-to-use API allowing you to build your own server-side plugins ranging from existing legacy applications written COBOL to modern CORBA or COM/OLE based object-oriented systems written in for instance C++ or Smalltalk." ORCL000762-763. COM and OLE are object-handling extensions, as I understand that term to be used in the Patents	Not asserted.
14: The computer-implemented method in claim 13 wherein said object handling extension is an OLE extension	See '335 13. OLE is taught expressly	Not asserted.
16[15(a)]: A computer-	OWS 2.0 is computer-implemented	Dr. Shamos did not prove that References 604 and 605

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
implemented method comprising the steps of:		meet the preamble.
16[15(b)(1)]: transferring a request from an HTTP-compliant device to a page server	The Web server is Oracle's Web listener. The page server is a "WRB Executable Engine." "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), a program to be accessed through the CGI interface, or whether access to the file system of the machine on which the Listener resides is sufficient. If WRB access is required, the Listener passes the request to WRB Dispatcher for processing." p. 1-2. The WRB sends the request to an appropriate page server. Hence the required routing is performed	Under Oracle's proposed construction, the "page server" is "page generating software, running on a processor separate from that of the HTTP-compliant device, that generates dynamic Web pages." Dr. Shamos failed to prove that Reference 604/605 discloses a page server under Oracle's proposed construction because he failed to identify a "page server" that executes on a processor separate from the processor of the HTTP-compliant device. Dr. Shamos failed to identify an "HTTP-compliant device" in References 604/605.
16[15(b)(2)]: said page server receiving said request and	The request must be received to be processed	
16[15(b)(3)]: releasing said HTTP-compliant device to process other requests	"AII inter-process communication is handled by the transport independent WRB protocol. Webserver 2.0 supports standard IPC mechanisms" Technical Note, p. 5. A standard IPC mechanism is Socket-Bind-Listen, hence SBL releasing is performed according to EpicRealm's contention	Dr. Shamos contended that the "SBL Theory" of Releasing disclosed by References 604/605 meets this limitation under either party's construction. I agree. Furthermore, the fact that Web Server 2.0 supports a standard IPC mechanism, does not necessarily mean that socket-bind-listen is used for communications.
16[15(c)(1)]: wherein said transferring step further includes the steps of: intercepting said request at said HTTP-compliant device and	"The HTTP engine does not make any attempt at interpreting the URL, instead it is immediately handed off to the Web Request Broker for further processing." Technical Note, p. 5. This "handing off" without processing is intercepting.	References 604 and 605 do not disclose "intercepting" for Oracle 2.0. As the quote cited by Dr. Shamos demonstrates, there is no decision by the "Web server" whether it should process a request or whether a "page server" should process a request. When "no suitable WRB Service is found, the request is passed back to the Web Listener and standard processing continues." (Ex. 48, Technical Note, p. 5).

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
		These References teach away from "intercepting" in the claimed invention. Specifically, the Oracle WebServer 2.0 Technical Note states: "[t]he typical HTTP API enables a custom extension to be linked into the HTTP server process and allows it to intercept and fulfill a client request." (Ex. 48, Technical Note, p. 3) However, Oracle 2.0 does not implement "the typical HTTP API" which allows intercepting because "the problem with the HTTP API approach is that all extensions are linked into the same executable and thus share the same memory, process, and signal handling space. … Oracle's answer to these issues is a new and improved architecture for the server, which effectively layers Oracle Web Server 2.0 into two fundamental components: a HTTP engine (the Oracle Web Listener) and a high-speed dispatch mechanism (the Web Request Broker) which routes requests to server extensions running in separate processes." Thus, not only do these References not disclose "intercepting" for Oracle 2.0, they teach away from it. Furthermore, Dr. Shamos failed to analyze these References under either epicRealm's or Oracle's proposed constructions.
16[15(c)(2)]: transferring said request to said page server	The Web server is Oracle's Web listener. The page server is a "WRB Executable Engine." "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), a program to be accessed through the CGI interface, or whether access to the file system of the machine on which the Listener resides is sufficient. If WRB access is required, the Listener passes the request to	

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
	WRB Dispatcher for processing." p. 1-2. The WRB sends the request to an appropriate page server. Hence the required routing is performed	
16[15(d)]: processing said request, said processing being performed by said page server while said HTTP-compliant device concurrently processes said other requests, and	Concurrency is taught expressly: "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), a program to be accessed through the CGI interface, or whether access to the file system of the machine on which the Listener resides is sufficient. If WRB access is required, the Listener passes the request to WRB Dispatcher for processing: then it returns to the task of listening for more incoming HTTP requests." p. 1-2. "Oracle's HTTP Server is designed to be a very scalable, high-performance network daemon. It is implemented as a multi-threaded, single-process asynchronous engine, allowing multiple concurrent requests to be processed at the same time using standard HTTP or HTTP over SSL" Technical Note, p. 3	
16[15(e)(1)]: dynamically generating a page in response to said request	"As documented in the CGI 1.1 specification, the Oracle Web Listener takes the contents of standard output and returns it to the Web browser that requested the dynamic HTML document." p. 6-6	
16[15(e)(2)]: said page including data dynamically retrieved from one or more data sources	"The Oracle WebServer ... draws on information from the database and the operating system's (0S) file system as necessary to respond to the request. The file system can be used for static (hardcoded) Web pages, or form scripts that do not access the database, and the database is used for Web pages that are	

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
	generated at runtime using "live" data." p. 1-2	
16(a): The computer-implemented method in claim 15 wherein said step of transferring said request includes the steps of:	OWS 2.0 is computer-implemented	Dr. Shamos did not prove that References 604 and 605 meet the preamble.
16(b): transferring said request from said HTTP-compliant device to a dispatcher, and	The dispatcher is the "WRB Dispatcher." "Whenever the Web Listener receives a URL that calls for the WRB. it passes execution of the request to the WRB Dispatcher or simply Dispatcher. The Dispatcher maintains communication with a pool of processes called WRB Executable Engines (WRBXs).The Dispatcher finds a free WRBX that is configured to run the desired WRB Service and passes execution to it." p. 1-11	As discussed in the section below, Dr. Shamos failed to establish that References 604 and 605 have a "dispatcher" because References 604 and 605 do not disclose "dispatching" under either Oracle's or epicRealm's proposed construction.
16(c): dispatching said request to said page server	The WRB Dispatcher makes an informed selection of which page server (WRBX) should process the request based on dynamic information: which WRBX is free and which one is configured to run the desired service. Load balancing is also performed: "WRB Executable Engine (WRBX). One of a pool of processes that the WRB maintains continuously, so that HTTP requests requiring the execution of programs are not slowed down by the performance cost of spawning a new process. WRBX's are associated with WRB Cartridges and are created and destroyed according to workload." p. A-13. "The WRB dispatcher must decide what type of object is being requested. To do this, it examines the WRB configuration file, which maps virtual directories to WRB Services. ... The WRB Dispatcher performs	References 604 and 605 do not disclose "dispatching" under epicRealm's or Oracle's construction. EpicRealm's proposed construction for "dispatching" is "examining a request to make an informed selection of which page server should process the request based on dynamic information maintained about page servers, the dynamic information indicating which page server can more efficiently process the request, and sending the request to the selected page server." References 604/605 do not disclose maintaining dynamic information indicating which page server can more efficiently process the request nor do References 604/605 disclose making an informed selection of which page server should process the request. The information used by the WRB dispatcher only indicates

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
	dynamic load balancing between multiple instances of a WRB Service, and the webmaster can configure each WRB Service to have a minimum and maximum number of instances. An instance corresponds to a process in a UNIX environment." Technical Note, p. 5	which WRBXs are executing a request and which are free to process other requests. A busy WRBX (one already processing a request) cannot process another request. The dispatcher does not maintain information about which one of the free WRBXs can more efficiently process a request. Thus, these References do not disclose maintaining dynamic information indicating which page server can more efficiently process the request. Dr. Shamos argued that the maintained dynamic information is "which WRBX is free and which one is configured to run the desired service." As discussed above, information about which WRBX is free is not information about which page server can more efficiently process a request. Further, information about which WRBX is configured to run a desired service is not dynamic information because the configuration information is stored in a static configuration file. (See also my response to Dr. Shamos's "Configuration File Dispatching.") Similarly, the information about which WRBX is configured to run a desired service is not information about which page server can more efficiently process a request because it does not differentiate between the relative efficiencies of page servers. A WRBX that is not configured to run a desired service is not an available page server to process a request, thus it is not considered in determining which page server can more efficiently process a request. In addition, these References do not disclose making an informed selection of which page server should process a request based on dynamic information. It is my

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
		understanding that the WRB dispatcher sends a request to a free WRBX that can process the request. This decision is not "informed" because it does not distinguish between the free page servers that can process a request. Similarly, Dr. Shamos also failed to prove that these References disclose "dispatching" under Oracle's construction. At a minimum, Dr. Shamos did not distinguish his analysis under epicRealm's and Oracle's proposed constructions for "dispatching." Therefore, for the reasons set forth above, these References do not disclose "dispatching." In addition, Oracle's construction requires that the "dispatcher" make an "informed selection of which page server should process the request…." In these References, there is not an "informed selection" for the reasons set forth above. Thus, these References do not disclose "dispatching" under Oracle's proposed construction. Dr. Shamos contended that these references disclose "load balancing" but failed to explain what relevance his "load balancing" has to the claim terms. Indeed, there are many forms of load balancing that do not meet the limitations of the claims.
17: The computer-implemented method in claim 15 wherein said step of processing said request includes the step of identifying said one or more data sources from which to retrieve said data	See '554 1b6 (configuration file)	Not asserted.

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
18: The computer-implemented method in claim 15 wherein said step of dynamically generating said Web page includes the step of dynamically retrieving said data from said one or more data sources	"PL/SQL cartridges. These execute stored PL/SQL procedures to generate HTML dynamically using Oracle data." p. 1-3. "If desired, you can use LiveHTML to run the PL/SOL Agent under CGI. and thereby incorporate dynamic Oracle data in hardcoded Web pages." p. 1-17	Not asserted.
19: The computer-implemented method in claim 15 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more data sources	"To do this, the Web Listener passes requests intended for these programs to the WRB Dispatcher, which maintains a pool of processes to which it can assign the requests." p. 1-7. This "pool of processes" is a connection cache	Not asserted.
20: The computer-implemented method in claim 15 wherein said step of processing said request includes the step of logging into said one or more data sources	"A URL that invokes the PL/SQL Agent must specify a DCD (Database Connection Descriptor). This is an OS file maintained by the WebServer that provides the username, password, database, and other information to be used to establish the database connection." p. 1-14. "At the lowest level, Oracle; internal API allows us t o establish a connection to an Oracle7 Server using SQL*Net in one step, and actually logging on to the database in a separate step." Technical Note, p. 11	Not asserted.
21: The computer-implemented method in claim 15 wherein said step of dynamically generating said page includes the step of maintaining a page cache	"Page caching. Ordinarily, when the Web Listener opens a requested file, the file remains open and mapped into memory until all clients using the file are finished with it, at which point the Web Listener closes the file and frees the memory mapping associated with it. The Web Listener allows you to	Not asserted.

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
containing said page	specify files to be cached. Cached files are opened when a client requests them, and remain open for the life of the Web Listener process. This optimizes access times for files, such as your home page, that are requested often." p. 1-5	
22: The computer-implemented method in claim 15 wherein said page server includes tag-based text templates for configuring said page	LiveHTML comprises tag-based text templates. See '554 7	Not asserted.
23: The computer-implemented method in claim 22 wherein said step of processing said request further includes the step of inserting said dynamically retrieved data from said one or more data sources into said tag-based text templates	See '554 8.	Not asserted.
24: The computer-implemented method in claim 22 wherein at least one of said tag-based text templates drives a format of the data dynamically retrieved from said one or more data sources in response to said request	Section 5 (e.g., page 5-5, MAC003147) describes the use of tag-based text templates in PL/SQL procedures to drive the format of the data in creating the generated page. "The Oracle WebServer Developer's Toolkit is a set of procedures that help the customer create programs to generate dynamic documents that are easily presented in HTML." p. 1-2 (MAC003052). HTML determines the format of a web page	Not asserted.
25: The computer-implemented method in claim	See '554 8.	Not asserted.

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)	Dr. Finkel's Analysis
22 wherein said tag-based text templates include HTML templates		
26: The computer-implemented method in claim 15 wherein said step of processing said request further includes the step of dynamically updating data at said one or more data sources	"You can also use a cursor to update a table's contents. To do this, use a FOR UPDATE clause to lock the rows in the table. See "Using FOR UPDATE" in the PL/SQL User's Guide and Reference for more information. Sometimes, you may need to use cursor variables. which are not associated with a query until runtime. This is a form of dynamic SQL." p. B-19	Not asserted.
27: The computer-implemented method in claim 15 wherein said step of processing said request further includes the step of processing an object handling extension	"The Java Interpreter includes classes for native Oracle7 access via PL/SQL and dynamic creation of HTML documents. For customers who wish to employ their own programming language of choice, Oracle WebServer 2.0 offers an easy-to-use API allowing you to build your own server-side plugins ranging from existing legacy applications written COBOL to modern CORBA or COM/OLE based object-oriented systems written in for instance C++ or Smalltalk." ORCL000762-763. COM and OLE are object-handling extensions, as I understand that term to be used in the Patents	Not asserted.
28: The computer-implemented method in claim 27 wherein said object handling extension is an OLE extension	See '335 13. OLE is taught expressly	Not asserted.
29(a): A computer-implemented method comprising the steps of:	OWS 2.0 is computer-implemented	Not asserted.

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
29(b): transferring a request from an HTTP-compliant device to a dispatcher	The dispatcher is the "WRB Dispatcher." "Whenever the Web Listener receives a URL that calls for the WRB. it passes execution of the request to the WRB Dispatcher or simply Dispatcher. The Dispatcher maintains communication with a pool of processes called WRB Executable Engines (WRBXs).The Dispatcher finds a free WRBX that is configured to run the desired WRB Service and passes execution to it." p. 1-11	Not asserted.
29(c): maintaining dynamic information regarding data sources a given page server may access	See '554 1b6	Not asserted.
29(d)(1): dispatching said request to an appropriate page server based on said request and based on said dynamic information	The WRB Dispatcher makes an informed selection of which page server (WRBX) should process the request based on dynamic information: which WRBX is free and which one is configured to run the desired service. Load balancing is also performed: "WRB Executable Engine (WRBX). One of a pool of processes that the WRB maintains continuously, so that HTTP requests requiring the execution of programs are not slowed down by the performance cost of spawning a new process. WRBX's are associated with WRB Cartridges and are created and destroyed according to workload." p. A-13. "The WRB dispatcher must decide what type of object is being requested. To do this, it examines the WRB configuration file, which maps virtual directories to WRB Services. ... The WRB Dispatcher performs dynamic load balancing between multiple instances of a WRB Service, and the webmaster can configure each WRB Service to have a minimum and maximum	Not asserted.

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
	number of instances. An instance corresponds to a process in a UNIX environment." Technical Note, p. 5	
29(d)(2): said page server receiving said request and	The request must be received to be processed	Not asserted.
29(d)(3): releasing said HTTP-compliant device to process other requests	"AII inter-process communication is handled by the transport independent WRB protocol. Webserver 2.0 supports standard IPC mechanisms" Technical Note, p. 5. A standard IPC mechanism is Socket-Bind-Listen, hence SBL releasing is performed according to EpicRealm's contention	Not asserted.
29(e): processing said request, said processing being performed by said page server while said HTTP-compliant device concurrently processes said other requests, and	Concurrency is taught expressly: "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), a program to be accessed through the CGI interface, or whether access to the file system of the machine on which the Listener resides is sufficient. If WRB access is required, the Listener passes the request to WRB Dispatcher for processing: then it returns to the task of listening for more incoming HTTP requests." p. 1-2. "Oracle's HTTP Server is designed to be a very scalable, high-performance network daemon. It is implemented as a multi-threaded, single-process asynchronous engine, allowing multiple concurrent requests to be processed at the same time using standard HTTP or HTTP over SSL" Technical Note, p. 3	Not asserted.
29(f)(1): dynamically generating a Web page in	"As documented in the CGI 1.1 specification, the Oracle Web Listener takes the contents of standard	Not asserted.

Oracle WebServer User's Guide Release 2.0.2 Production, Part No. A23646-2 (604/605)

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly From His Claim Charts)**	**Dr. Finkel's Analysis**
response to said request	output and returns it to the Web browser that requested the dynamic HTML document." p. 6-6	
29(f)(2): said page including data dynamically retrieved from one or more data sources	"The Oracle WebServer ... draws on information from the database and the operating system's (0S) file system as necessary to respond to the request. The file system can be used for static (hardcoded) Web pages, or form scripts that do not access the database, and the database is used for Web pages that are generated at runtime using "live" data." p. 1-2	Not asserted.

Exhibit 10

Oracle WebServer User's Guide Release 2.0 Production, Part No. A23646-1 and Oracle WebServer 2.0 Technical Note

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
1(a): A computer-implemented method for managing a dynamic Web page generation request to a Web server, said computer-implemented method comprising the steps of:	"Using the Oracle Web Listener, your web site can respond to client requests by generating HTML documents dynamically." p. 1-7. The "database is used for Web pages that are generated at runtime using "live" data." p. 1-2	Dr. Shamos did not prove that these References meet the preamble. In addition, my response to Dr. Shamos's section on Oracle Developments sets forth further reasons why these References do not anticipate any claims of the patents-in-suit.
1(b)(1): routing said request from said Web server to a page server	"The HTTP Server is the network layer component of the Oracle WebServer. It listens for incoming HTTP requests, delivers static files and runs simple CGI programs and hands off everything else to the Web Request Broker (WRB), where a request is handled by a server extension." ORCL 000764. "The WRB dispatcher must decide what type of object is being requested. To do this, it examines the WRB configuration file, which maps virtual directories to WRB Services. For instance, the Web Request Broker may be configured to dispatch all URL requests beginning with /java to the integrated Java Interpreter. However, if no suitable WRB Service is found, the request is passed back to the Web Listener and standard processing continues." ORCL000766.	Under Oracle's proposed construction, the "page server" is "page generating software, running on a processor separate from that of the Web server, that generates dynamic Web pages." Dr. Shamos failed to prove that these references disclose a page server under Oracle's proposed construction because he failed to identify a "page server" that executes on a processor separate from the processor of the Web Server.
1(b)(2): said page server receiving said request and	The request must be received to be processed. "The PL/SQL Agent performs two major tasks which enable a stored procedure to generate a dynamic HTML page. The first is to launch the procedure and pass in any parameters that were part of the HTTP request." ORCL000771.	
1(b)(3): releasing said Web server to process other requests	"AII inter-process communication is handled by the transport independent WRB protocol. Webserver 2.0 supports standard IPC mechanisms" Technical Note, p. 5. A standard IPC mechanism is Socket-Bind-Listen, hence SBL releasing is performed according to EpicRealm's contention. The Web Request Broker API document	Dr. Shamos contended that the "SBL theory" of releasing disclosed by these Reference meets this limitation under either party's construction. I agree.

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
	has a diagram at page ORCL01606867 that shows that WEB Application (such as PL/SQL Agent) sends REQ COMPLETE Response message, which is an act that is separate from merely receiving the request. HTTP Server receives the REQ COMPLETE Response message and frees up certain resources that would either allow processing of other requests or facilitate more efficient processing of already received requests. HTTP Server must free such resources because otherwise the REQ COMPLETE Response message has no purpose. ORCL01606867. Under the SBL theory, the INIT Response message sent from Web Application (such as PL/SQL Agent) to HTTP Server is an action, which is separate from merely receiving the request that informs the HTTP Server that it is now able to offload the processing of the request to the now initialized WEB Application, prospectively freeing up the HTTP Server to process other requests once a request capable of being processed by the WEB Application is received.	Further, the "Web Request Broker API" is a separate Reference not incorporated by either of the References relied on by Dr. Shamos. Thus, it should not be considered as part of this anticipatory reference. Moreover, the Web Request Broker API does not distinguish whether the REQ Complete Response relied on by Dr. Shamos is sent to the WRB Dispatcher (Dr. Shamos's Dispatcher) or the HTTP Server (Dr. Shamos's Web Server). In addition, the fact that Web Server 2.0 supports a standard IPC mechanism, does not necessarily mean that socket-bind-listen is used for communications.
1(b)(4): wherein said routing step further includes the steps of intercepting said request at said Web server	"The HTTP engine does not make any attempt at interpreting the URL, instead it is immediately handed off to the Web Request Broker for further processing." Technical Note, p. 5. This "handing off" without processing is intercepting.	These References do not disclose "intercepting" for Oracle 2.0. As the quote cited by Dr. Shamos demonstrates, there is no decision by the "Web server" whether it should process a request or whether a "page server" should process a request. When "no suitable DRB Service is found, the request is passed back to the Web Listener and standard processing continues." (Ex. 48, Technical Note, p. 5) These References teach away from "intercepting" in the claimed invention. Specifically, the Oracle WebServer 2.0 Technical Note states: "[t]he typical HTTP API enables a custom extension to be linked

Oracle WebServer User's Guide Release 2.0 Production, Part No. A23646-1 and Oracle WebServer 2.0 Technical Note

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
		into the HTTP server process and allows it to intercept and fulfill a client request." (Ex. 48, p. 3) However, Oracle 2.0 does not implement the "the typical HTTP API" which allows intercepting because "the problem with the HTTP API approach is that all extensions are linked into the same executable and thus share the same memory, process, and signal handling space. … Oracle's answer to these issues is a new and improved architecture for the server, which effectively layers Oracle Web Server 2.0 into two fundamental components: a HTTP engine (the Oracle Web Listener) and a high-speed dispatch mechanism (the Web Request Broker) which routes requests to server extensions running in separate processes." Thus, not only do these References not disclose "intercepting" for Oracle 2.0, they teach away from it. Furthermore, Dr. Shamos failed to analyze these References under either epicRealm's or Oracle's proposed constructions.
1(b)(5): routing said request from said Web server to a dispatcher, and	"The HTTP Server is the network layer component of the Oracle WebServer. It listens for incoming HTTP requests, delivers static files and runs simple CGI programs and hands off everything else to the Web Request Broker (WRB), where a request is handled by a server extension." ORCL 000764.	As discussed in the section below, Dr. Shamos failed to establish that these References have a "dispatcher" because these References do not disclose "dispatching" under either Oracle's or epicRealm's proposed construction.
1(b)(6): dispatching said request to said page server	The WRB Dispatcher makes an informed selection of which page server (WRBX) should process the request based on dynamic information: which WRBX is free and which one is configured to run	These References do not disclose "dispatching" under epicRealm's or Oracle's construction.

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
	the desired service. Load balancing is also performed: "WRB Executable Engine (WRBX). One of a pool of processes that the WRB maintains continuously, so that HTTP requests requiring the execution of programs are not slowed down by the performance cost of spawning a new process. WRBX's are associated with WRB Cartridges and are created and destroyed according to workload." p. A-13. "The WRB dispatcher must decide what type of object is being requested. To do this, it examines the WRB configuration file, which maps virtual directories to WRB Services. ... The WRB Dispatcher performs dynamic load balancing between multiple instances of a WRB Service, and the webmaster can configure each WRB Service to have a minimum and maximum number of instances. An instance corresponds to a process in a UNIX environment." Technical Note, p. 5	EpicRealm's proposed construction for "dispatching" is "examining a request to make an informed selection of which page server should process the request based on dynamic information maintained about page servers, the dynamic information indicating which page server can more efficiently process the request, and sending the request to the selected page server." These References do not disclose maintaining dynamic information indicating which page server can more efficiently process the request nor do these References disclose making an informed selection of which page server should process the request. The dynamic information maintained by the WRB dispatcher only indicates which WRBXs are executing requests and which are free to process other requests. A busy WRBX cannot process a request. The dispatcher does not maintain information about which one of the free WRBXs can more efficiently process a request. Thus, these References do not disclose maintaining dynamic information indicating which page server can more efficiently process the request. Dr. Shamos argued that the maintained dynamic information is "which WRBX is free and which one is configured to run the desired service." As discussed above, information about which WRBX is free is not information

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
		about which page server can more efficiently process a request because a busy WRBX cannot process a request. Further, information about which WRBX is configured to run a desired service is not dynamic information because the configuration information is stored in a static configuration file. Similarly, the information about which WRBX is configured to run a desired service is not information about which page server can more efficiently process a request because it does not differentiate the between the relative efficiencies of page servers. A WRBX that is not configured to run a desired service is not an available page server to process a request, thus it is not considered in determining whether a page server can more efficiently process a request. In addition, these References do not disclose making an informed selection of which page server should process a request based on that dynamic information. It is my understanding that the WRB Dispatcher sends a request to a free WRBX to process the request. The WRB Dispatcher's selection of a "page server" is not informed because it does not distinguish between the free page servers that can process a request. Similarly, Dr. Shamos also failed to demonstrate that these References disclose "dispatching" under Oracle's construction.

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
		At a minimum, Dr. Shamos did not distinguish his analysis under epicRealm's and Oracle's proposed construction for "dispatching." Therefore, for the reasons set forth above, these References do not disclose "dispatching." In addition, Oracle's construction requires that the "dispatcher" make an "informed selection of which page server should process the request…." In these References, there is not an "informed selection" for the reasons set forth above. Thus, these References do not disclose "dispatching" under Oracle's proposed construction. Dr. Shamos contended that these references disclose "load balancing" but fails to explain what relevance the "load balancing" has to the claim terms. Indeed, there are many forms of load balancing that do not meet the limitations of the claims, including the form of "load balancing" disclosed by these References for the reasons set forth above.
1(c): processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests, and	Concurrency is taught expressly: "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), a program to be accessed through the CGI interface, or whether access to the file system of the machine on which the Listener resides is sufficient. If WRB access is required, the Listener passes the request to WRB Dispatcher for processing: then it returns to the task of listening for more incoming HTTP requests." p. 1-2. "Oracle's HTTP Server is designed to be a very scalable, high-performance network daemon. It is implemented as a multi-threaded, single-process asynchronous	

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)**	**Dr. Finkel's Analysis**
	engine, allowing multiple concurrent requests to be processed at the same time using standard HTTP or HTTP over SSL" Technical Note, p. 3	
1(d)(1): dynamically generating a Web page in response to said request	"As documented in the CGI 1.1 specification, the Oracle Web Listener takes the contents of standard output and returns it to the Web browser that requested the dynamic HTML document." p. 6-6. "QUERY_FORM prints an HTML page with all the columns for the specified table. Invoke the procedure from a Web Browser with a URL like: http://yourhost:port_num/service_name/owa/query_form?the_table=emp . . . Invoking this procedure brings up a page that looks like [page shown at ORCL01816677]. Also see ORCL01816678-79.	
1(d)(2): said Web page including data dynamically retrieved from one or more data sources.	"The Oracle WebServer ... draws on information from the database and the operating system's (0S) file system as necessary to respond to the request. The file system can be used for static (hardcoded) Web pages, or form scripts that do not access the database, and the database is used for Web pages that are generated at runtime using "live" data." p. 1-2. Also: "LiveHTML code is formatted as SGML comments, so that it is ignored should the file ever find its way t o the browser unparsed. The format for LiveHTML codes, therefore, is the following: <!--#command tag1="value1" tag2="value2" --> The tags are arguments to the commands, most of which actually only accept one of the possible tags. The possible commands and their associated tags are as follows: . . . include. This command specifies that a file is to be included in the generated HTML page at this point." ORCL000778-779.	
2: The computer-implemented method in claim 1 wherein said step of processing said	See '554 1b6 (configuration file). "[PL/SQL Agent] also needs to have access to all the details required to actually establish a permanent connection: typically a set of environment variables (ORACLE_HOME and ORACLE_SID for UNIX systems) and	These References cannot anticipate this claim because they do not anticipate the claim from which this claim depends.

Oracle WebServer User's Guide Release 2.0 Production, Part No. A23646-1 and Oracle WebServer 2.0 Technical Note

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
request includes the step of identifying said one or more data sources from which to retrieve said data.	perhaps a SQL*Net connect descriptor. All of this information is referred to as a DCD (Database Connection Descriptor), and all of the DCDs known to the PL/SQL Agent are stored in a configuration file named owa.cfg (DCDs where known as Web Agent Services in Webserver 1.0). . . . DCDs are defined in the PL/SQL Agent's configuration file, but they are activated by creating a directory mapping in the Web Request Broker's configuration file." ORCL000772.	Dr. Shamos failed to demonstrate how these References meet Oracle's construction for this claim. Oracle construes "identifying said one or more data sources" to mean "listing in an HTML extension tag the location of said one or more data sources." Dr. Shamos failed to demonstrate how this construction is met.
3: The computer-implemented method in claim 2 wherein said step of dynamically generating said Web page includes the step of dynamically retrieving said data from said one or more data sources.	Oracle WebServer User's Guide Release 2.0.2 Production, at page ORCL000833 discloses a web page that contains data retrieved in response to user's request. In particular, column headings and data in the table has been retrieved from a database table. "LiveHTML code is formatted as SGML comments, so that it is ignored should the file ever find its way t o the browser unparsed. The format for LiveHTML codes, therefore, is the following: <!--#command tag1="value1" tag2="value2" --> The tags are arguments to the commands, most of which actually only accept one of the possible tags. The possible commands and their associated tags are as follows: . . . include. This command specifies that a file is t o be included in the generated HTML page at this point." ORCL000778-779.	These References cannot anticipate this claim because they do not anticipate the claim from which this claim depends.
4: The computer-implemented method in claim 3 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more data sources.	"To do this, the Web Listener passes requests intended for these programs to the WRB Dispatcher, which maintains a pool of processes to which it can assign the requests." p. 1-7. This "pool of processes" is a connection cache	These References cannot anticipate this claim because they do not anticipate the claim from which this claim depends. Dr. Shamos failed to demonstrate that these References disclose "maintaining a connection cache to said one or more data sources." Indeed, Dr. Shamos failed to identify the one or more data sources required by this claim. Further, a pool of processes is not a "connection cache" as one skilled in the

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)**	**Dr. Finkel's Analysis**
		art would understand that term.
5: The computer-implemented method in claim 3 wherein said step of processing said request includes the step of logging into said one or more data sources.	"A URL that invokes the PL/SQL Agent must specify a DCD (Database Connection Descriptor). This is an OS file maintained by the WebServer that provides the username, password, database, and other information to be used to establish the database connection." p. 1-14. "At the lowest level, Oracle; internal API allows us t o establish a connection to an Oracle7 Server using SQL*Net in one step, and actually logging on to the database in a separate step." Technical Note, p. 11	These References cannot anticipate this claim because they do not anticipate the claim from which this claim depends.
6: The computer-implemented method in claim 3 wherein said step of dynamically generating said Web page includes the step of maintaining a page cache containing said Web page.	"Page caching. Ordinarily, when the Web Listener opens a requested file, the file remains open and mapped into memory until all clients using the file are finished with it, at which point the Web Listener closes the file and frees the memory mapping associated with it. The Web Listener allows you to specify files to be cached. Cached files are opened when a client requests them, and remain open for the life of the Web Listener process. This optimizes access times for files, such as your home page, that are requested often." p. 1-5	Not asserted.
7: The computer-implemented method in claim 3 wherein said page server includes custom HTML extension templates for configuring said Web page.	"LiveHTML. A way to embed dynamic content in Web pages. This content can be either other Web pages or the output of scripts run by the Operating System. LiveHTML is an Oracle extension of the NCSA standard Server Side Includes functionality." p. 1-2. "The LiveHTML Interpreter enables you to include dynamic content in otherwise static Web pages" p. 1-16. "LiveHTML code is formatted as SGML comments ... The format for LiveHTML codes, therefore, is the following: < ! -- #command tag1="value1" tag2="value2" -- >. The tags are arguments to the commands, most of which actually only accept one of the possible tags." p. 4-30. "LiveHTML files supplement HTML with instructions that WebServer executes before transmitting the page. These instructions specify material that is to be included in the generated page. Said material can include other Web pages, environment variables, and the output of programs executed	These References cannot anticipate this claim because they do not anticipate the claim from which this claim depends.

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)**	**Dr. Finkel's Analysis**
	on the Server." p. A-8	
8: The computer-implemented method in claim 7 wherein said step of processing said request further includes the step of inserting said dynamically retrieved data from said one or more data sources into said custom HTML extension templates.	"include. This command specifies that a file is to be included in the generated HTML page at this point. The file can any of the following: - another LiveHTML file like the current one. - a regular HTML file. - an ASCII file." p. 4-30	These References cannot anticipate this claim because they do not anticipate the claim from which this claim depends.
9(a): A networked system for managing a dynamic Web page generation request, said system comprising	OWS 2.0 operates with a Web server, hence is networked	Dr. Shamos did not prove that these References meet the preamble. In addition, my response to Dr. Shamos's section on Oracle Developments sets forth further reasons why these References do not anticipate any claims of the patents-in-suit.
9(b): one or more data sources	"The Oracle WebServer ... draws on information from the database and the operating system's (0S) file system as necessary to respond to the request. The file system can be used for static (hardcoded) Web pages, or form scripts that do not access the database, and the database is used for Web pages that are generated at runtime using "live" data." p. 1-2. Also: "LiveHTML code is formatted as SGML comments, so that it is ignored should the file ever find its way t o the browser unparsed. The format for LiveHTML codes, therefore, is the following: <!--#command tag1="value1" tag2="value2" --> The tags are arguments t o the commands, most of which actually only accept one of the possible tags. The possible commands and their associated tags are as follows: . . . include. This command specifies that a file is t o be included in the generated HTML page at this	

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)**	**Dr. Finkel's Analysis**
	point." ORCL000778-779.	
9(c): a page server having a processing means	"The HTTP Server is the network layer component of the Oracle WebServer. It listens for incoming HTTP requests, delivers static files and runs simple CGI programs and hands off everything else to the Web Request Broker (WRB), where a request is handled by a server extension." ORCL 000764. "The WRB dispatcher must decide what type of object is being requested. To do this, it examines the WRB configuration file, which maps virtual directories to WRB Services. For instance, the Web Request Broker may be configured to dispatch all URL requests beginning with /java to the integrated Java Interpreter. However, if no suitable WRB Service is found, the request is passed back to the Web Listener and standard processing continues." ORCL000766. Thus the WRB services are "page servers" of the patent. A processor of a computer that runs WRB services is the page server processing means.	Under Oracle's proposed construction, the "page server" is "page generating software, running on a processor separate from that of the second computer system, that generates dynamic Web pages." Dr. Shamos failed to prove that these references disclose a page server under Oracle's proposed construction because he failed to identify a "page server" that executes on a processor separate from the processor of the second computer system. Dr. Shamos failed to identify the specified function and the corresponding structure, material or acts for the "processing means" as required by 35 U.S.C. § 112 ¶ 6. Furthermore, Dr. Shamos failed to demonstrate that these References disclose the specified function and corresponding structure, material or acts for the "processing means" or equivalents thereof.
9(d): a first computer system including means for generating said request, and	The first computer is a computer that has a processor that runs a web browser as shown on page 1-3. See ORCL000763. "From a web browser's point of view, Oracle WebServer 2.0 looks just like any other web server." Technical Note, p. 2. Therefore, means for generating said request is a processor of a computer running a web browser as show in the figure at ORCL000763.	Dr. Shamos failed to identify the specified function and the corresponding structure, material or acts for the "means for generating said request" as required by 35 U.S.C. § 112 ¶ 6. Furthermore, Dr. Shamos failed to demonstrate that these References disclose the specified function and corresponding structure, material or acts for the "means for generating said request" or equivalents thereof.

Oracle WebServer User's Guide Release 2.0 Production, Part No. A23646-1 and Oracle WebServer 2.0 Technical Note

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)**	**Dr. Finkel's Analysis**
9(e)(1): a second computer system including	The second computer system includes the Web listener, the Web Request Broker, as shown on Technical Note p. 2, and the WRB Executable Engines (page servers)	
9(e)(2): means for receiving said request from said first computer	"The HTTP Server is the network layer component of the Oracle WebServer. It listens for incoming HTTP requests, delivers static files and runs simple CGI programs and hands off everything else to the Web Request Broker (WRB), where a request is handled by a server extension." ORCL 000764. "The WRB dispatcher must decide what type of object is being requested. To do this, it examines the WRB configuration file, which maps virtual directories to WRB Services. For instance, the Web Request Broker may be configured to dispatch all URL requests beginning with /java to the integrated Java Interpreter. However, if no suitable WRB Service is found, the request is passed back to the Web Listener and standard processing continues." ORCL000766. Means for receiving said request is the processor of a computer that runs the Web Listener. See figure on ORCL000763.	Dr. Shamos failed to identify the specified function and the corresponding structure, material or acts for the "means for receiving" as required by 35 U.S.C. § 112 ¶ 6. Furthermore, Dr. Shamos failed to demonstrate that these References disclose the specified function and corresponding structure, material or acts for the "means for receiving" or equivalents thereof.
9(e)(3): said second computer system also including a router, said router routing said request from said second computer system to said page server	"The HTTP Server is the network layer component of the Oracle WebServer. It listens for incoming HTTP requests, delivers static files and runs simple CGI programs and hands off everything else to the Web Request Broker (WRB), where a request is handled by a server extension." ORCL 000764. "The WRB dispatcher must decide what type of object is being requested. To do this, it examines the WRB configuration file, which maps virtual directories to WRB Services. For instance, the Web Request Broker may be configured to dispatch all URL requests beginning with /java to the integrated Java Interpreter. However, if no suitable WRB Service is found, the request is passed back to the Web Listener and standard processing continues." ORCL000766. Also: the executable code that performs the routing at the second computer system is the router.	

Oracle WebServer User's Guide Release 2.0 Production, Part No. A23646-1 and Oracle WebServer 2.0 Technical Note

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)**	**Dr. Finkel's Analysis**
9(e)(4): wherein said routing further includes intercepting said request at said second computer	"The HTTP engine does not make any attempt at interpreting the URL, instead it is immediately handed off to the Web Request Broker for further processing." Technical Note, p. 5. This "handing off" without processing is intercepting.	These References do not disclose "intercepting" for Oracle 2.0. As the quote cited by Dr. Shamos demonstrates, there is no decision by the "Web server" whether it should process a request or whether a "page server" should process a request. When "no suitable DRB Service is found, the request is passed back to the Web Listener and standard processing continues." (Ex. 48, Technical Note, p. 5) These References teach away from "intercepting" in the claimed invention. Specifically, the Oracle WebServer 2.0 Technical Note states: "[t]he typical HTTP API enables a custom extension to be linked into the HTTP server process and allows it to intercept and fulfill a client request." (Ex. 48, Technical Note, p. 3) However, Oracle 2.0 does not implement the "the typical HTTP API" which allows intercepting because "the problem with the HTTP API approach is that all extensions are linked into the same executable and thus share the same memory, process, and signal handling space. … Oracle's answer to these issues is a new and improved architecture for the server, which effectively layers Oracle Web Server 2.0 into two fundamental components: a HTTP engine (the Oracle Web Listener) and a high-speed dispatch mechanism (the Web Request Broker) which routes requests to server extensions running in separate processes."

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)**	**Dr. Finkel's Analysis**
		Thus, not only do these References not disclose "intercepting" for Oracle 2.0, they teach away from it. Furthermore, Dr. Shamos failed to analyze these References under either epicRealm's or Oracle's proposed constructions.
9(e)(5): routing said request from said second computer to a dispatcher, and	"The HTTP Server is the network layer component of the Oracle WebServer. It listens for incoming HTTP requests, delivers static files and runs simple CGI programs and hands off everything else to the Web Request Broker (WRB), where a request is handled by a server extension." ORCL 000764.	As discussed in the section below, Dr. Shamos failed to establish that these References have a "dispatcher" because these References do not disclose "dispatching" under either Oracle's or epicRealm's proposed construction.
9(e)(6): dispatching said request to said page server	The WRB Dispatcher makes an informed selection of which page server (WRBX) should process the request based on dynamic information: which WRBX is free and which one is configured to run the desired service. Load balancing is also performed: "WRB Executable Engine (WRBX). One of a pool of processes that the WRB maintains continuously, so that HTTP requests requiring the execution of programs are not slowed down by the performance cost of spawning a new process. WRBX's are associated with WRB Cartridges and are created and destroyed according to workload." p. A-13. "The WRB dispatcher must decide what type of object is being requested. To do this, it examines the WRB configuration file, which maps virtual directories to WRB Services. ... The WRB Dispatcher performs dynamic load balancing between multiple instances of a WRB Service, and the webmaster can configure each WRB Service to have a minimum and maximum number of instances. An instance corresponds to a process in a UNIX environment." Technical Note, p. 5	These References do not disclose "dispatching" under epicRealm's or Oracle's construction. EpicRealm's proposed construction for "dispatching" is "examining a request to make an informed selection of which page server should process the request based on dynamic information maintained about page servers, the dynamic information indicating which page server can more efficiently process the request, and sending the request to the selected page server." These References do not disclose maintaining dynamic information indicating which page server can more efficiently process the request nor do these References disclose making an informed selection of which page server should process the request. The dynamic information maintained by the WRB dispatcher only

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
		indicates which WRBXs are executing requests and which are free to process other requests. A busy WRBX cannot process a request. The dispatcher does not maintain information about which one of the free WRBXs can more efficiently process a request. Thus, these References do not disclose maintaining dynamic information indicating which page server can more efficiently process the request. Dr. Shamos argued that the maintained dynamic information is "which WRBX is free and which one is configured to run the desired service." As discussed above, information about which WRBX is free is not information about which page server can more efficiently process a request because a busy WRBX cannot process a request. Further, information about which WRBX is configured to run a desired service is not dynamic information because the configuration information is stored in a static configuration file. Similarly, the information about which WRBX is configured to run a desired service is not information about which page server can more efficiently process a request because it does not differentiate the between the relative efficiencies of page servers. A WRBX that is not configured to run a desired service is not an available page server to process a request, thus it is not considered in determining whether a page

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
		server can more efficiently process a request. In addition, these References do not disclose making an informed selection of which page server should process a request based on that dynamic information. It is my understanding that the WRB Dispatcher sends a request to a free WRBX to process the request. The WRB Dispatcher's selection of a "page server" is not informed because it does not distinguish between the free page servers that can process a request. Similarly, Dr. Shamos also failed to demonstrate that these References disclose "dispatching" under Oracle's construction. At a minimum, Dr. Shamos did not distinguish his analysis under epicRealm's and Oracle's proposed construction for "dispatching." Therefore, for the reasons set forth above, these References do not disclose "dispatching." In addition, Oracle's construction requires that the "dispatcher" make an "informed selection of which page server should process the request…." In these References, there is not an "informed selection" for the reasons set forth above. Thus, these References do not disclose "dispatching" under Oracle's proposed construction. Dr. Shamos contended that these references disclose "load balancing" but fails to explain

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
		what relevance the "load balancing" has to the claim terms. Indeed, there are many forms of load balancing that do not meet the limitations of the claims, including the form of "load balancing" disclosed by these References for the reasons set forth above.
9(e)(7): said page server receiving said request and	The request must be received to be processed. "The PL/SQL Agent performs two major tasks which enable a stored procedure to generate a dynamic HTML page. The first is to launch the procedure and pass in any parameters that were part of the HTTP request." ORCL000771.	
9(e)(8): releasing said second computer system to process other requests	"AII inter-process communication is handled by the transport independent WRB protocol. Webserver 2.0 supports standard IPC mechanisms" Technical Note, p. 5. A standard IPC mechanism is Socket-Bind-Listen, hence SBL releasing is performed according to EpicRealm's contention. The Web Request Broker API document has a diagram at page ORCL01606867 that shows that WEB Application (such as PL/SQL Agent) sends REQ COMPLETE Response message, which is an act that is separate from merely receiving the request. HTTP Server receives the REQ COMPLETE Response message and frees up certain resources that would either allow processing of other requests or facilitate more efficient processing of already received requests. HTTP Server must free such resources because otherwise the REQ COMPLETE Response message has no purpose. ORCL01606867. Under the SBL theory, the INIT Response message sent from Web Application (such as PL/SQL Agent) to HTTP Server is an action, which is separate from merely receiving the request that informs the HTTP Server that it is now able to offload the processing of the request to the now initialized WEB Application, prospectively freeing up the HTTP Server to process other requests once a request capable of being processed by the WEB Application is received.	Dr. Shamos contended that the "SBL theory" of releasing disclosed by these Reference meets this limitation under either party's construction. I agree. Further, the "Web Request Broker API" is a separate Reference not incorporated by either of the References relied on by Dr. Shamos. Thus, it should not be considered as part of this anticipatory reference. Moreover, the Web Request Broker API does not distinguish whether the REQ Complete Response relied on by Dr. Shamos is sent to the WRB Dispatcher (Dr. Shamos's Dispatcher) or the HTTP Server (Dr. Shamos's Web Server). In addition, the fact that Web Server 2.0 supports a standard IPC mechanism, does not necessarily mean that socket-bind-listen is used for communications.

Oracle WebServer User's Guide Release 2.0 Production, Part No. A23646-1 and Oracle WebServer 2.0 Technical Note

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
9(e)(9): said page server processing means processing said request and dynamically generating a Web page in response to said request	"As documented in the CGI 1.1 specification, the Oracle Web Listener takes the contents of standard output and returns it to the Web browser that requested the dynamic HTML document." p. 6-6. "QUERY_FORM prints an HTML page with all the columns for the specified table. Invoke the procedure from a Web Browser with a URL like: http://yourhost:port_num/service_name/owa/query_form?the_table=emp . . . Invoking this procedure brings up a page that looks like [page shown at ORCL01816677]. Also see ORCL01816678-79.	
9(e)(10): said Web page including data dynamically retrieved from said one or more data sources	"The Oracle WebServer ... draws on information from the database and the operating system's (0S) file system as necessary to respond to the request. The file system can be used for static (hardcoded) Web pages, or form scripts that do not access the database, and the database is used for Web pages that are generated at runtime using "live" data." p. 1-2. Also: "LiveHTML code is formatted as SGML comments, so that it is ignored should the file ever find its way t o the browser unparsed. The format for LiveHTML codes, therefore, is the following: <!--#command tag1="value1" tag2="value2" --> The tags are arguments t o the commands, most of which actually only accept one of the possible tags. The possible commands and their associated tags are as follows: . . . include. This command specifies that a file is t o be included in the generated HTML page at this point." ORCL000778-779.	
10(a): The networked system in claim 9 wherein said router in said second computer system includes:	OWS 2.0 operates with a Web server, hence is networked	These References cannot anticipate this claim because they do not anticipate the claim from which this claim depends.
10(b): an interceptor	"The HTTP engine does not make any attempt at interpreting the	These References cannot anticipate this claim

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)**	**Dr. Finkel's Analysis**
intercepting said request at said second computer system and routing said request, and	URL, instead it is immediately handed off to the Web Request Broker for further processing." Technical Note, p. 5. This "handing off" without processing is intercepting.	because they do not anticipate the claim from which this claim depends. In addition, my response to Dr. Shamos's section on Oracle Developments sets forth further reasons why these References do not anticipate any claims of the patents-in-suit. These References do not disclose "intercepting" for Oracle 2.0. As the quote cited by Dr. Shamos demonstrates, there is no decision by the "Web server" whether it should process a request or whether a "page server" should process a request. When "no suitable DRB Service is found, the request is passed back to the Web Listener and standard processing continues." (Ex. 48, Technical Note, p. 5) These References teach away from "intercepting" in the claimed invention. Specifically, the Oracle WebServer 2.0 Technical Note states: "[t]he typical HTTP API enables a custom extension to be linked into the HTTP server process and allows it to intercept and fulfill a client request." (Ex. 48, Technical Note, p. 3) However, Oracle 2.0 does not implement the "the typical HTTP API" which allows intercepting because "the problem with the HTTP API approach is that all extensions are linked into the same executable and thus share the same memory, process, and signal handling space. …

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
		Oracle's answer to these issues is a new and improved architecture for the server, which effectively layers Oracle Web Server 2.0 into two fundamental components: a HTTP engine (the Oracle Web Listener) and a high-speed dispatch mechanism (the Web Request Broker) which routes requests to server extensions running in separate processes." Thus, not only do these References not disclose "intercepting" for Oracle 2.0, they teach away from it. Furthermore, Dr. Shamos failed to analyze these References under either epicRealm's or Oracle's proposed constructions.
10(c): a dispatcher receiving said routed request from said interceptor and dispatching said request to said page server	The WRB Dispatcher makes an informed selection of which page server (WRBX) should process the request based on dynamic information: which WRBX is free and which one is configured to run the desired service. Load balancing is also performed: "WRB Executable Engine (WRBX). One of a pool of processes that the WRB maintains continuously, so that HTTP requests requiring the execution of programs are not slowed down by the performance cost of spawning a new process. WRBX's are associated with WRB Cartridges and are created and destroyed according to workload." p. A-13. "The WRB dispatcher must decide what type of object is being requested. To do this, it examines the WRB configuration file, which maps virtual directories to WRB Services. ... The WRB Dispatcher performs dynamic load balancing between multiple instances of a WRB Service, and the webmaster can configure each WRB Service to have a minimum and maximum number of instances. An instance corresponds to a process in a UNIX environment." Technical Note, p. 5	These References cannot anticipate this claim because they do not anticipate the claim from which this claim depends. Dr. Shamos failed to establish that these References have a "dispatcher" because these References do not disclose "dispatching" under either Oracle's or epicRealm's proposed construction. These References do not disclose "dispatching" under epicRealm's or Oracle's construction. EpicRealm's proposed construction for "dispatching" is "examining a request to make an informed selection of which page server should process the request based on

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
		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." These References do not disclose maintaining dynamic information indicating which page server can more efficiently process the request nor do these References disclose making an informed selection of which page server should process the request. The dynamic information maintained by the WRB dispatcher only indicates which WRBXs are executing requests and which are free to process other requests. A busy WRBX cannot process a request. The dispatcher does not maintain information about which one of the free WRBXs can more efficiently process a request. Thus, these References do not disclose maintaining dynamic information indicating which page server can more efficiently process the request. Dr. Shamos argued that the maintained dynamic information is "which WRBX is free and which one is configured to run the desired service." As discussed above, information about which WRBX is free is not information about which page server can more efficiently process a request because a busy WRBX cannot process a request. Further, information about which WRBX is configured to run a desired service is not

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
		dynamic information because the configuration information is stored in a static configuration file. Similarly, the information about which WRBX is configured to run a desired service is not information about which page server can more efficiently process a request because it does not differentiate the between the relative efficiencies of page servers. A WRBX that is not configured to run a desired service is not an available page server to process a request, thus it is not considered in determining whether a page server can more efficiently process a request. In addition, these References do not disclose making an informed selection of which page server should process a request based on that dynamic information. It is my understanding that the WRB Dispatcher sends a request to a free WRBX to process the request. The WRB Dispatcher's selection of a "page server" is not informed because it does not distinguish between the free page servers that can process a request. Similarly, Dr. Shamos also failed to demonstrate that these References disclose "dispatching" under Oracle's construction. At a minimum, Dr. Shamos did not distinguish his analysis under epicRealm's and Oracle's proposed construction for "dispatching." Therefore, for the reasons set forth above, these References do not disclose

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
		"dispatching." In addition, Oracle's construction requires that the "dispatcher" make an "informed selection of which page server should process the request…." In these References, there is not an "informed selection" for the reasons set forth above. Thus, these References do not disclose "dispatching" under Oracle's proposed construction. Dr. Shamos contended that these references disclose "load balancing" but fails to explain what relevance the "load balancing" has to the claim terms. Indeed, there are many forms of load balancing that do not meet the limitations of the claims, including the form of "load balancing" disclosed by these References for the reasons set forth above.
11(a): A machine readable medium having stored thereon data representing sequences of instructions, which when executed by a computer system, cause said computer system to perform the steps of:	OWS 2.0 is a programmed system whose instructions are on a machine readable medium	Dr. Shamos only contended that these References disclose "dynamically generating a Web page in response to said request." Dr. Shamos failed to contend that these References disclose "dynamically generating a Web page, said Web page including data retrieved from one or more data sources." Dr. Shamos did not prove that these References meet the preamble. In addition, my response to Dr. Shamos's section on Oracle Developments sets forth further reasons why these References do not anticipate any claims of the patents-in-suit.

Oracle WebServer User's Guide Release 2.0 Production, Part No. A23646-1 and Oracle WebServer 2.0 Technical Note

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
11(b)(1): routing a dynamic Web page generation request from a Web server to a page server	"The HTTP Server is the network layer component of the Oracle WebServer. It listens for incoming HTTP requests, delivers static files and runs simple CGI programs and hands off everything else to the Web Request Broker (WRB), where a request is handled by a server extension." ORCL 000764. "The WRB dispatcher must decide what type of object is being requested. To do this, it examines the WRB configuration file, which maps virtual directories to WRB Services. For instance, the Web Request Broker may be configured to dispatch all URL requests beginning with /java to the integrated Java Interpreter. However, if no suitable WRB Service is found, the request is passed back to the Web Listener and standard processing continues." ORCL000766.	Under Oracle's proposed construction, the "page server" is "page generating software, running on a processor separate from that of the Web server, that generates dynamic Web pages." Dr. Shamos failed to prove that these references disclose a page server under Oracle's proposed construction because he failed to identify a "page server" that executes on a processor separate from the processor of the Web Server.
11(b)(2): said page server receiving said request and	The request must be received to be processed. "The PL/SQL Agent performs two major tasks which enable a stored procedure to generate a dynamic HTML page. The first is to launch the procedure and pass in any parameters that were part of the HTTP request." ORCL000771.	
11(b)(3): releasing said Web server to process other requests	"AII inter-process communication is handled by the transport independent WRB protocol. Webserver 2.0 supports standard IPC mechanisms" Technical Note, p. 5. A standard IPC mechanism is Socket-Bind-Listen, hence SBL releasing is performed according to EpicRealm's contention. The Web Request Broker API document has a diagram at page ORCL01606867 that shows that WEB Application (such as PL/SQL Agent) sends REQ COMPLETE Response message, which is an act that is separate from merely receiving the request. HTTP Server receives the REQ COMPLETE Response message and frees up certain resources that would either allow processing of other requests or facilitate more efficient processing of already received requests. HTTP Server must free such resources because otherwise the REQ COMPLETE Response message has no purpose. ORCL01606867. Under the SBL theory,	Dr. Shamos contended that the "SBL theory" of releasing disclosed by these Reference meets this limitation under either party's construction. I agree. Further, the "Web Request Broker API" is a separate Reference not incorporated by either of the References relied on by Dr. Shamos. Thus, it should not be considered as part of this anticipatory reference. Moreover, the Web Request Broker API does not distinguish whether the REQ Complete Response relied on by Dr. Shamos is sent to the WRB Dispatcher (Dr. Shamos's Dispatcher) or the

'554 Claim Language	**Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)**	**Dr. Finkel's Analysis**
	the INIT Response message sent from Web Application (such as PL/SQL Agent) to HTTP Server is an action, which is separate from merely receiving the request that informs the HTTP Server that it is now able to offload the processing of the request to the now initialized WEB Application, prospectively freeing up the HTTP Server to process other requests once a request capable of being processed by the WEB Application is received.	HTTP Server (Dr. Shamos's Web Server). In addition, the fact that Web Server 2.0 supports a standard IPC mechanism, does not necessarily mean that socket-bind-listen is used for communications.
11(b)(4): wherein said routing step further includes the steps of intercepting said request at said Web server	"The HTTP engine does not make any attempt at interpreting the URL, instead it is immediately handed off to the Web Request Broker for further processing." Technical Note, p. 5. This "handing off" without processing is intercepting.	These References do not disclose "intercepting" for Oracle 2.0. As the quote cited by Dr. Shamos demonstrates, there is no decision by the "Web server" whether it should process a request or whether a "page server" should process a request. When "no suitable DRB Service is found, the request is passed back to the Web Listener and standard processing continues." (Ex. 48, Technical Note, p. 5) These References teach away from "intercepting" in the claimed invention. Specifically, the Oracle WebServer 2.0 Technical Note states: "[t]he typical HTTP API enables a custom extension to be linked into the HTTP server process and allows it to intercept and fulfill a client request." (Ex. 48, Technical Note, p. 3) However, Oracle 2.0 does not implement the "the typical HTTP API" which allows intercepting because "the problem with the HTTP API approach is that all extensions are linked into the same executable and thus share the same memory, process, and signal handling space. …

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
		Oracle's answer to these issues is a new and improved architecture for the server, which effectively layers Oracle Web Server 2.0 into two fundamental components: a HTTP engine (the Oracle Web Listener) and a high-speed dispatch mechanism (the Web Request Broker) which routes requests to server extensions running in separate processes." Thus, not only do these References not disclose "intercepting" for Oracle 2.0, they teach away from it. Furthermore, Dr. Shamos failed to analyze these References under either epicRealm's or Oracle's proposed constructions.
11(b)(5): routing said request from said Web server to a dispatcher, and	"The HTTP Server is the network layer component of the Oracle WebServer. It listens for incoming HTTP requests, delivers static files and runs simple CGI programs and hands off everything else to the Web Request Broker (WRB), where a request is handled by a server extension." ORCL 000764.	As discussed in the section below, Dr. Shamos failed to establish that these References have a "dispatcher" because these References do not disclose "dispatching" under either Oracle's or epicRealm's proposed construction.
11(b)(6): dispatching said request to said page server	The WRB Dispatcher makes an informed selection of which page server (WRBX) should process the request based on dynamic information: which WRBX is free and which one is configured to run the desired service. Load balancing is also performed: "WRB Executable Engine (WRBX). One of a pool of processes that the WRB maintains continuously, so that HTTP requests requiring the execution of programs are not slowed down by the performance cost of spawning a new process. WRBX's are associated with WRB Cartridges and are created and destroyed according to workload." p. A-13. "The WRB dispatcher must decide what type of object is being requested. To do this, it examines the WRB configuration file, which maps virtual directories to WRB Services. ... The WRB	These References do not disclose "dispatching" under epicRealm's or Oracle's construction. EpicRealm's proposed construction for "dispatching" is "examining a request to make an informed selection of which page server should process the request based on dynamic information maintained about page servers, the dynamic information indicating which page server can more efficiently process the request, and sending the request to

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
	Dispatcher performs dynamic load balancing between multiple instances of a WRB Service, and the webmaster can configure each WRB Service to have a minimum and maximum number of instances. An instance corresponds to a process in a UNIX environment." Technical Note, p. 5	the selected page server." These References do not disclose maintaining dynamic information indicating which page server can more efficiently process the request nor do these References disclose making an informed selection of which page server should process the request. The dynamic information maintained by the WRB dispatcher only indicates which WRBXs are executing requests and which are free to process other requests. A busy WRBX cannot process a request. The dispatcher does not maintain information about which one of the free WRBXs can more efficiently process a request. Thus, these References do not disclose maintaining dynamic information indicating which page server can more efficiently process the request. Dr. Shamos argued that the maintained dynamic information is "which WRBX is free and which one is configured to run the desired service." As discussed above, information about which WRBX is free is not information about which page server can more efficiently process a request because a busy WRBX cannot process a request. Further, information about which WRBX is configured to run a desired service is not dynamic information because the configuration information is stored in a static configuration file. Similarly, the information about which WRBX is configured to run a

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
		desired service is not information about which page server can more efficiently process a request because it does not differentiate the between the relative efficiencies of page servers. A WRBX that is not configured to run a desired service is not an available page server to process a request, thus it is not considered in determining whether a page server can more efficiently process a request. In addition, these References do not disclose making an informed selection of which page server should process a request based on that dynamic information. It is my understanding that the WRB Dispatcher sends a request to a free WRBX to process the request. The WRB Dispatcher's selection of a "page server" is not informed because it does not distinguish between the free page servers that can process a request. Similarly, Dr. Shamos also failed to demonstrate that these References disclose "dispatching" under Oracle's construction. At a minimum, Dr. Shamos did not distinguish his analysis under epicRealm's and Oracle's proposed construction for "dispatching." Therefore, for the reasons set forth above, these References do not disclose "dispatching." In addition, Oracle's construction requires that the "dispatcher" make an "informed selection of which page server should process the request…." In these

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
		References, there is not an "informed selection" for the reasons set forth above. Thus, these References do not disclose "dispatching" under Oracle's proposed construction. Dr. Shamos contended that these references disclose "load balancing" but fails to explain what relevance the "load balancing" has to the claim terms. Indeed, there are many forms of load balancing that do not meet the limitations of the claims, including the form of "load balancing" disclosed by these References for the reasons set forth above.
11(c): processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests, and	Concurrency is taught expressly: "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), a program to be accessed through the CGI interface, or whether access to the file system of the machine on which the Listener resides is sufficient. If WRB access is required, the Listener passes the request to WRB Dispatcher for processing: then it returns to the task of listening for more incoming HTTP requests." p. 1-2. "Oracle's HTTP Server is designed to be a very scalable, high-performance network daemon. It is implemented as a multi-threaded, single-process asynchronous engine, allowing multiple concurrent requests to be processed at the same time using standard HTTP or HTTP over SSL" Technical Note, p. 3	
11(d): dynamically generating a Web page, said Web page including data retrieved from one or more data sources	"As documented in the CGI 1.1 specification, the Oracle Web Listener takes the contents of standard output and returns it to the Web browser that requested the dynamic HTML document." p. 6-6. "QUERY_FORM prints an HTML page with all the columns for the specified table. Invoke the procedure from a Web Browser with a	Dr. Shamos only contended that these References disclose "dynamically generating a Web page in response to said request." Dr. Shamos failed to contend that these References disclose "dynamically generating

Oracle WebServer User's Guide Release 2.0 Production, Part No. A23646-1 and Oracle WebServer 2.0 Technical Note

'554 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
	URL like: http://yourhost:port_num/service_name/owa/query_form?the_table=emp . . . Invoking this procedure brings up a page that looks like [page shown at ORCL01816677]. Also see ORCL01816678-79.	a Web page, said Web page including data retrieved from one or more data sources."

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)**	**Dr. Finkel's Analysis**
2[1(a)]: A computer-implemented method for managing a dynamic Web page generation request to a Web server, said computer-implemented method comprising the steps of:	"Using the Oracle Web Listener, your web site can respond to client requests by generating HTML documents dynamically." p. 1-7. The "database is used for Web pages that are generated at runtime using "live" data." p. 1-2	In addition, my response to Dr. Shamos's section on Oracle Developments sets forth further reasons why these References do not anticipate any claims of the patents-in-suit.
2[1(b)(1)]: routing a request from a Web server to a page server	"The HTTP Server is the network layer component of the Oracle WebServer. It listens for incoming HTTP requests, delivers static files and runs simple CGI programs and hands off everything else to the Web Request Broker (WRB), where a request is handled by a server extension." ORCL 000764. "The WRB dispatcher must decide what type of object is being requested. To do this, it examines the WRB configuration file, which maps virtual directories to WRB Services. For instance, the Web Request Broker may be configured to dispatch all URL requests beginning with /java to the integrated Java Interpreter. However, if no suitable WRB Service is found, the request is passed back to the Web Listener and standard processing continues." ORCL000766.	Under Oracle's proposed construction, the "page server" is "page generating software, running on a processor separate from that of the Web server, that generates dynamic Web pages." Dr. Shamos failed to prove that these references disclose a page server under Oracle's proposed construction because he failed to identify a "page server" that executes on a processor separate from the processor of the Web Server.
2[1(b)(2)]: said page server receiving said request and	The request must be received to be processed. "The PL/SQL Agent performs two major tasks which enable a stored procedure to generate a dynamic HTML page. The first is to launch the procedure and pass in any parameters that were part of the HTTP request." ORCL000771.	
2[1(b)(3)]: releasing said Web server to process other requests	"AII inter-process communication is handled by the transport independent WRB protocol. Webserver 2.0 supports standard IPC mechanisms" Technical Note, p. 5. A standard IPC mechanism is Socket-Bind-Listen, hence SBL releasing is performed according to EpicRealm's contention. The Web Request Broker API document	Dr. Shamos contended that the "SBL theory" of releasing disclosed by these Reference meets this limitation under either party's construction. I agree.

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
	has a diagram at page ORCL01606867 that shows that WEB Application (such as PL/SQL Agent) sends REQ COMPLETE Response message, which is an act that is separate from merely receiving the request. HTTP Server receives the REQ COMPLETE Response message and frees up certain resources that would either allow processing of other requests or facilitate more efficient processing of already received requests. HTTP Server must free such resources because otherwise the REQ COMPLETE Response message has no purpose. ORCL01606867. Under the SBL theory, the INIT Response message sent from Web Application (such as PL/SQL Agent) to HTTP Server is an action, which is separate from merely receiving the request that informs the HTTP Server that it is now able to offload the processing of the request to the now initialized WEB Application, prospectively freeing up the HTTP Server to process other requests once a request capable of being processed by the WEB Application is received.	Further, the "Web Request Broker API" is a separate Reference not incorporated by either of the References relied on by Dr. Shamos. Thus, it should not be considered as part of this anticipatory reference. Moreover, the Web Request Broker API does not distinguish whether the REQ Complete Response relied on by Dr. Shamos is sent to the WRB Dispatcher (Dr. Shamos's Dispatcher) or the HTTP Server (Dr. Shamos's Web Server). In addition, the fact that Web Server 2.0 supports a standard IPC mechanism, does not necessarily mean that socket-bind-listen is used for communications.
2[1(c)(1)]: wherein said routing step further includes the steps of: intercepting said request at said Web server and	"The HTTP engine does not make any attempt at interpreting the URL, instead it is immediately handed off to the Web Request Broker for further processing." Technical Note, p. 5. This "handing off" without processing is intercepting.	These References do not disclose "intercepting" for Oracle 2.0. As the quote cited by Dr. Shamos demonstrates, there is no decision by the "Web server" whether it should process a request or whether a "page server" should process a request. When "no suitable DRB Service is found, the request is passed back to the Web Listener and standard processing continues." (Ex. 48, Technical Note, p. 5) These References teach away from "intercepting" in the claimed invention. Specifically, the Oracle WebServer 2.0 Technical Note states: "[t]he typical HTTP API enables a custom extension to be linked

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
		into the HTTP server process and allows it to intercept and fulfill a client request." (Ex. 48, Technical Note, p. 3) However, Oracle 2.0 does not implement the "the typical HTTP API" which allows intercepting because "the problem with the HTTP API approach is that all extensions are linked into the same executable and thus share the same memory, process, and signal handling space. … Oracle's answer to these issues is a new and improved architecture for the server, which effectively layers Oracle Web Server 2.0 into two fundamental components: a HTTP engine (the Oracle Web Listener) and a high-speed dispatch mechanism (the Web Request Broker) which routes requests to server extensions running in separate processes." Thus, not only do these References not disclose "intercepting" for Oracle 2.0, they teach away from it. Furthermore, Dr. Shamos failed to analyze these References under either epicRealm's or Oracle's proposed constructions.
2[1(c)(2)]: routing said request to said page server	"The HTTP Server is the network layer component of the Oracle WebServer. It listens for incoming HTTP requests, delivers static files and runs simple CGI programs and hands off everything else to the Web Request Broker (WRB), where a request is handled by a server extension." ORCL 000764. "The WRB dispatcher must decide what type of object is being requested. To do this, it examines the WRB configuration file, which maps virtual directories to WRB Services. For instance, the Web Request Broker may be configured to dispatch all URL requests beginning with	

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
	/java to the integrated Java Interpreter. However, if no suitable WRB Service is found, the request is passed back to the Web Listener and standard processing continues." ORCL000766.	
2[1(d)]: processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests, and	Concurrency is taught expressly: "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), a program to be accessed through the CGI interface, or whether access to the file system of the machine on which the Listener resides is sufficient. If WRB access is required, the Listener passes the request to WRB Dispatcher for processing: then it returns to the task of listening for more incoming HTTP requests." p. 1-2. "Oracle's HTTP Server is designed to be a very scalable, high-performance network daemon. It is implemented as a multi-threaded, single-process asynchronous engine, allowing multiple concurrent requests to be processed at the same time using standard HTTP or HTTP over SSL" Technical Note, p. 3	
2[1(e)(1)]: dynamically generating a Web page in response to said request	"As documented in the CGI 1.1 specification, the Oracle Web Listener takes the contents of standard output and returns it to the Web browser that requested the dynamic HTML document." p. 6-6. "QUERY_FORM prints an HTML page with all the columns for the specified table. Invoke the procedure from a Web Browser with a URL like: http://yourhost:port_num/service_name/owa/query_form?the_table=emp . . . Invoking this procedure brings up a page that looks like [page shown at ORCL01816677]. Also see ORCL01816678-79.	
2[1(e)(2)]: said Web page including data dynamically retrieved from one or more data sources	"The Oracle WebServer ... draws on information from the database and the operating system's (0S) file system as necessary to respond to the request. The file system can be used for static (hardcoded) Web pages, or form scripts that do not access the database, and the database is used for Web pages that are generated at runtime using	

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
	"live" data." p. 1-2. Also: "LiveHTML code is formatted as SGML comments, so that it is ignored should the file ever find its way t o the browser unparsed. The format for LiveHTML codes, therefore, is the following: <!--#command tag1="value1" tag2="value2" --> The tags are arguments t o the commands, most of which actually only accept one of the possible tags. The possible commands and their associated tags are as follows: . . . include. This command specifies that a file is t o be included in the generated HTML page at this point." ORCL000778-779.	
2(a): The computer-implemented method in claim 1 wherein said step of routing said request includes the steps of:	"Using the Oracle Web Listener, your web site can respond to client requests by generating HTML documents dynamically." p. 1-7. The "database is used for Web pages that are generated at runtime using "live" data." p. 1-2	Dr. Shamos did not prove that these References meet the preamble. In addition, my response to Dr. Shamos's section on Oracle Developments sets forth further reasons why these References do not anticipate any claims of the patents-in-suit.
2(b): routing said request from said Web server to a dispatcher, and	"The HTTP Server is the network layer component of the Oracle WebServer. It listens for incoming HTTP requests, delivers static files and runs simple CGI programs and hands off everything else to the Web Request Broker (WRB), where a request is handled by a server extension." ORCL 000764.	As discussed in the section below, Dr. Shamos failed to establish that these References have a "dispatcher" because these References do not disclose "dispatching" under either Oracle's or epicRealm's proposed construction.
2(c): dispatching said request to said page server	The WRB Dispatcher makes an informed selection of which page server (WRBX) should process the request based on dynamic information: which WRBX is free and which one is configured to run the desired service. Load balancing is also performed: "WRB Executable Engine (WRBX). One of a pool of processes that the WRB maintains continuously, so that HTTP requests requiring the execution of programs are not slowed down by the performance cost of spawning a new process. WRBX's are associated with WRB Cartridges and are created and destroyed according to workload." p. A-13. "The WRB dispatcher must decide what type of object is	These References do not disclose "dispatching" under epicRealm's or Oracle's construction. EpicRealm's proposed construction for "dispatching" is "examining a request to make an informed selection of which page server should process the request based on dynamic information maintained about page servers, the dynamic information indicating

Oracle WebServer User's Guide Release 2.0 Production, Part No. A23646-1 and Oracle WebServer 2.0 Technical Note

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
	being requested. To do this, it examines the WRB configuration file, which maps virtual directories to WRB Services. ... The WRB Dispatcher performs dynamic load balancing between multiple instances of a WRB Service, and the webmaster can configure each WRB Service to have a minimum and maximum number of instances. An instance corresponds to a process in a UNIX environment." Technical Note, p. 5	which page server can more efficiently process the request, and sending the request to the selected page server." These References do not disclose maintaining dynamic information indicating which page server can more efficiently process the request nor do these References disclose making an informed selection of which page server should process the request. The dynamic information maintained by the WRB dispatcher only indicates which WRBXs are executing requests and which are free to process other requests. A busy WRBX cannot process a request. The dispatcher does not maintain information about which one of the free WRBXs can more efficiently process a request. Thus, these References do not disclose maintaining dynamic information indicating which page server can more efficiently process the request. Dr. Shamos argued that the maintained dynamic information is "which WRBX is free and which one is configured to run the desired service." As discussed above, information about which WRBX is free is not information about which page server can more efficiently process a request because a busy WRBX cannot process a request. Further, information about which WRBX is configured to run a desired service is not dynamic information because the configuration information is stored in a static

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
		configuration file. Similarly, the information about which WRBX is configured to run a desired service is not information about which page server can more efficiently process a request because it does not differentiate the between the relative efficiencies of page servers. A WRBX that is not configured to run a desired service is not an available page server to process a request, thus it is not considered in determining whether a page server can more efficiently process a request. In addition, these References do not disclose making an informed selection of which page server should process a request based on that dynamic information. It is my understanding that the WRB Dispatcher sends a request to a free WRBX to process the request. The WRB Dispatcher's selection of a "page server" is not informed because it does not distinguish between the free page servers that can process a request. Similarly, Dr. Shamos also failed to demonstrate that these References disclose "dispatching" under Oracle's construction. At a minimum, Dr. Shamos did not distinguish his analysis under epicRealm's and Oracle's proposed construction for "dispatching." Therefore, for the reasons set forth above, these References do not disclose "dispatching." In addition, Oracle's construction requires that the "dispatcher"

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)**	**Dr. Finkel's Analysis**
		make an "informed selection of which page server should process the request…." In these References, there is not an "informed selection" for the reasons set forth above. Thus, these References do not disclose "dispatching" under Oracle's proposed construction. Dr. Shamos contended that these references disclose "load balancing" but fails to explain what relevance the "load balancing" has to the claim terms. Indeed, there are many forms of load balancing that do not meet the limitations of the claims, including the form of "load balancing" disclosed by these References for the reasons set forth above.
3: The computer-implemented method in claim 1 wherein said step of processing said request includes the step of identifying said one or more data sources from which to retrieve said data	See '554 1b6 (configuration file). "[PL/SQL Agent] also needs to have access to all the details required to actually establish a permanent connection: typically a set of environment variables (ORACLE_HOME and ORACLE_SID for UNIX systems) and perhaps a SQL*Net connect descriptor. All of this information is referred to as a DCD (Database Connection Descriptor), and all of the DCDs known t o the PL/SQL Agent are stored in a configuration file named owa.cfg (DCDs where known as Web Agent Services in Webserver 1.0). . . . DCDs are defined in the PL/SQL Agent's configuration file, but they are activated by creating a directory mapping in the Web Request Broker's configuration file." ORCL000772.	Not asserted.
4: The computer-implemented method in claim 1 wherein said step of dynamically	Oracle WebServer User's Guide Release 2.0.2 Production, at page ORCL000833 discloses a web page that contains data retrieved in response to user's request. In particular, column headings and data in the table has been retrieved from a database table. "LiveHTML	Not asserted.

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
generating said Web page includes the step of dynamically retrieving said data from said one or more data sources	code is formatted as SGML comments, so that it is ignored should the file ever find its way t o the browser unparsed. The format for LiveHTML codes, therefore, is the following: <!--#command tag1="value1" tag2="value2" --> The tags are arguments to the commands, most of which actually only accept one of the possible tags. The possible commands and their associated tags are as follows: . . . include. This command specifies that a file is t o be included in the generated HTML page at this point." ORCL000778-779.	
5: The computer-implemented method in claim 1 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more data sources	"To do this, the Web Listener passes requests intended for these programs to the WRB Dispatcher, which maintains a pool of processes to which it can assign the requests." p. 1-7. This "pool of processes" is a connection cache	Not asserted.
6: The computer-implemented method in claim 1 wherein said step of processing said request includes the step of logging into said one or more data sources	"A URL that invokes the PL/SQL Agent must specify a DCD (Database Connection Descriptor). This is an OS file maintained by the WebServer that provides the username, password, database, and other information to be used to establish the database connection." p. 1-14. "At the lowest level, Oracle; internal API allows us t o establish a connection to an Oracle7 Server using SQL*Net in one step, and actually logging on to the database in a separate step." Technical Note, p. 11	Not asserted.
7: The computer-implemented method in claim 1 wherein said step of dynamically generating said Web page includes the step of	"Page caching. Ordinarily, when the Web Listener opens a requested file, the file remains open and mapped into memory until all clients using the file are finished with it, at which point the Web Listener closes the file and frees the memory mapping associated with it. The Web Listener allows you to specify files to be cached. Cached files are opened when a client requests them, and remain	Not asserted.

Oracle WebServer User's Guide Release 2.0 Production, Part No. A23646-1 and Oracle WebServer 2.0 Technical Note

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)**	**Dr. Finkel's Analysis**
maintaining a page cache containing said Web page	open for the life of the Web Listener process. This optimizes access times for files, such as your home page, that are requested often." p. 1-5	
8: The computer-implemented method in claim 1 wherein said page server includes tag-based text templates for configuring said Web page	LiveHTML comprises tag-based text templates. See '554 7	Not asserted.
9: The computer-implemented method in claim 8 wherein said step of processing said request further includes the step of inserting said-dynamically retrieved data from said one or more data sources into said tag-based text templates	See '554 8.	Not asserted.
10: The computer-implemented method in claim 8 wherein at least one of said tag-based text templates drives a format of the data dynamically retrieved from said one or more data sources in response to said request	!@@Section 5 (e.g., page 5-5, MAC003147) describes the use of tag-based text templates in PL/SQL procedures to drive the format of the data in creating the generated page. "The Oracle WebServer Developer's Toolkit is a set of procedures that help the customer create programs to generate dynamic documents that are easily presented in HTML." p. 1-2 (MAC003052). HTML determines the format of a web page	Not asserted.
11: The computer-implemented method in	See '554 8.	Not asserted.

Oracle WebServer User's Guide Release 2.0 Production, Part No. A23646-1 and Oracle WebServer 2.0 Technical Note

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)**	**Dr. Finkel's Analysis**
claim 8 wherein said tag-based text templates include HTML templates		
12: The computer-implemented method in claim 1 wherein said step of processing said request further includes the step of dynamically updating data at said one or more data sources	"You can also use a cursor to update a table's contents. To do this, use a FOR UPDATE clause to lock the rows in the table. See "Using FOR UPDATE" in the PL/SQL User's Guide and Reference for more information. Sometimes, you may need to use cursor variables. which are not associated with a query until runtime. This is a form of dynamic SQL." p. B-19. UPDATE is similar to SELECT in that it takes a predicate and operates on all rows that make the predicate TRUE. For example: UPDATE Customers SET ADDRESS = null WHERE LNAME = ' Subciak ' ORCL000972.	Not asserted.
13: The computer-implemented method in claim 1 wherein said step of processing said request further includes the step of processing an object handling extension	"The Java Interpreter includes classes for native Oracle7 access via PL/SQL and dynamic creation of HTML documents. For customers who wish to employ their own programming language of choice, Oracle WebServer 2.0 offers an easy-to-use API allowing you to build your own server-side plugins ranging from existing legacy applications written COBOL to modern CORBA or COM/OLE based object-oriented systems written in for instance C++ or Smalltalk." ORCL000762-763. COM and OLE are object-handling extensions, as I understand that term to be used in the Patents	Not asserted.
14: The computer-implemented method in claim 13 wherein said object handling extension is an OLE extension	See '335 13. OLE is taught expressly	Not asserted.
16[15(a)]: A computer-implemented method	OWS 2.0 is computer-implemented	Dr. Shamos did not prove that these References meet the preamble.

Oracle WebServer User's Guide Release 2.0 Production, Part No. A23646-1 and Oracle WebServer 2.0 Technical Note

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
comprising the steps of:		In addition, my response to Dr. Shamos's section on Oracle Developments sets forth further reasons why these References do not anticipate any claims of the patents-in-suit.
16[15(b)(1)]: transferring a request from an HTTP-compliant device to a page server	"The HTTP Server is the network layer component of the Oracle WebServer. It listens for incoming HTTP requests, delivers static files and runs simple CGI programs and hands off everything else to the Web Request Broker (WRB), where a request is handled by a server extension." ORCL 000764. "The WRB dispatcher must decide what type of object is being requested. To do this, it examines the WRB configuration file, which maps virtual directories to WRB Services. For instance, the Web Request Broker may be configured to dispatch all URL requests beginning with /java to the integrated Java Interpreter. However, if no suitable WRB Service is found, the request is passed back to the Web Listener and standard processing continues." ORCL000766.	Under Oracle's proposed construction, the "page server" is "page generating software, running on a processor separate from that of the HTTP-compliant device, that generates dynamic Web pages." Dr. Shamos failed to prove that these references disclose a page server under Oracle's proposed construction because he failed to identify a "page server" that executes on a processor separate from the processor of the Http-compliant device.
16[15(b)(2)]: said page server receiving said request and	The request must be received to be processed. "The PL/SQL Agent performs two major tasks which enable a stored procedure to generate a dynamic HTML page. The first is to launch the procedure and pass in any parameters that were part of the HTTP request." ORCL000771.	
16[15(b)(3)]: releasing said HTTP-compliant device to process other requests	"AII inter-process communication is handled by the transport independent WRB protocol. Webserver 2.0 supports standard IPC mechanisms" Technical Note, p. 5. A standard IPC mechanism is Socket-Bind-Listen, hence SBL releasing is performed according to EpicRealm's contention. The Web Request Broker API document has a diagram at page ORCL01606867 that shows that WEB Application (such as PL/SQL Agent) sends REQ COMPLETE Response message, which is an act that is separate from merely receiving the request. HTTP Server receives the REQ COMPLETE	Dr. Shamos contended that the "SBL theory" of releasing disclosed by these Reference meets this limitation under either party's construction. I agree. Further, the "Web Request Broker API" is a separate Reference not incorporated by either of the References relied on by Dr. Shamos. Thus, it should not be considered as part of

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
	Response message and frees up certain resources that would either allow processing of other requests or facilitate more efficient processing of already received requests. HTTP Server must free such resources because otherwise the REQ COMPLETE Response message has no purpose. ORCL01606867. Under the SBL theory, the INIT Response message sent from Web Application (such as PL/SQL Agent) to HTTP Server is an action, which is separate from merely receiving the request that informs the HTTP Server that it is now able to offload the processing of the request to the now initialized WEB Application, prospectively freeing up the HTTP Server to process other requests once a request capable of being processed by the WEB Application is received.	this anticipatory reference. Moreover, the Web Request Broker API does not distinguish whether the REQ Complete Response relied on by Dr. Shamos is sent to the WRB Dispatcher (Dr. Shamos's Dispatcher) or the HTTP Server (Dr. Shamos's Web Server). In addition, the fact that Web Server 2.0 supports a standard IPC mechanism, does not necessarily mean that socket-bind-listen is used for communications.
16[15(c)(1)]: wherein said transferring step further includes the steps of: intercepting said request at said HTTP-compliant device and	"The HTTP engine does not make any attempt at interpreting the URL, instead it is immediately handed off to the Web Request Broker for further processing." Technical Note, p. 5. This "handing off" without processing is intercepting.	These References do not disclose "intercepting" for Oracle 2.0. As the quote cited by Dr. Shamos demonstrates, there is no decision by the "Web server" whether it should process a request or whether a "page server" should process a request. When "no suitable DRB Service is found, the request is passed back to the Web Listener and standard processing continues." (Ex. 48, Technical Note, p. 5) These References teach away from "intercepting" in the claimed invention. Specifically, the Oracle WebServer 2.0 Technical Note states: "[t]he typical HTTP API enables a custom extension to be linked into the HTTP server process and allows it to intercept and fulfill a client request." (Ex. 48, Technical Note, p. 3) However, Oracle 2.0 does not implement the "the typical HTTP

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
		API" which allows intercepting because "the problem with the HTTP API approach is that all extensions are linked into the same executable and thus share the same memory, process, and signal handling space. … Oracle's answer to these issues is a new and improved architecture for the server, which effectively layers Oracle Web Server 2.0 into two fundamental components: a HTTP engine (the Oracle Web Listener) and a high-speed dispatch mechanism (the Web Request Broker) which routes requests to server extensions running in separate processes." Thus, not only do these References not disclose "intercepting" for Oracle 2.0, they teach away from it. Furthermore, Dr. Shamos failed to analyze these References under either epicRealm's or Oracle's proposed constructions.
16[15(c)(2)]: transferring said request to said page server	"The HTTP Server is the network layer component of the Oracle WebServer. It listens for incoming HTTP requests, delivers static files and runs simple CGI programs and hands off everything else to the Web Request Broker (WRB), where a request is handled by a server extension." ORCL 000764. "The WRB dispatcher must decide what type of object is being requested. To do this, it examines the WRB configuration file, which maps virtual directories to WRB Services. For instance, the Web Request Broker may be configured to dispatch all URL requests beginning with /java to the integrated Java Interpreter. However, if no suitable WRB Service is found, the request is passed back to the Web Listener and standard processing continues." ORCL000766.	Under Oracle's proposed construction, the "page server" is "a server machine, distinct from the [Web server machine/second computer system/HTTP-compliant device]…." Dr. Shamos failed to prove that these References disclose a page server under Oracle's proposed construction because he failed to identify a "page server" that is a server machine distinct from the "[Web server machine/second computer system/HTTP-compliant device]."

Oracle WebServer User's Guide Release 2.0 Production, Part No. A23646-1 and Oracle WebServer 2.0 Technical Note

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
16[15(d)]: processing said request, said processing being performed by said page server while said HTTP-compliant device concurrently processes said other requests, and	Concurrency is taught expressly: "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), a program to be accessed through the CGI interface, or whether access to the file system of the machine on which the Listener resides is sufficient. If WRB access is required, the Listener passes the request to WRB Dispatcher for processing: then it returns to the task of listening for more incoming HTTP requests." p. 1-2. "Oracle's HTTP Server is designed to be a very scalable, high-performance network daemon. It is implemented as a multi-threaded, single-process asynchronous engine, allowing multiple concurrent requests to be processed at the same time using standard HTTP or HTTP over SSL" Technical Note, p. 3	
16[15(e)(1)]: dynamically generating a page in response to said request	"As documented in the CGI 1.1 specification, the Oracle Web Listener takes the contents of standard output and returns it to the Web browser that requested the dynamic HTML document." p. 6-6. "QUERY_FORM prints an HTML page with all the columns for the specified table. Invoke the procedure from a Web Browser with a URL like: http://yourhost:port_num/service_name/owa/query_form?the_table=emp . . . Invoking this procedure brings up a page that looks like [page shown at ORCL01816677]. Also see ORCL01816678-79.	
16[15(e)(2)]: said page including data dynamically retrieved from one or more data sources	"The Oracle WebServer ... draws on information from the database and the operating system's (0S) file system as necessary to respond to the request. The file system can be used for static (hardcoded) Web pages, or form scripts that do not access the database, and the database is used for Web pages that are generated at runtime using "live" data." p. 1-2. Also: "LiveHTML code is formatted as SGML comments, so that it is ignored should the file ever find its way t o the browser unparsed. The format for LiveHTML codes,	

Oracle WebServer User's Guide Release 2.0 Production, Part No. A23646-1 and Oracle WebServer 2.0 Technical Note

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
	therefore, is the following: <!--#command tag1="value1" tag2="value2" --> The tags are arguments t o the commands, most of which actually only accept one of the possible tags. The possible commands and their associated tags are as follows: . . . include. This command specifies that a file is t o be included in the generated HTML page at this point." ORCL000778-779.	
16(a): The computer-implemented method in claim 15 wherein said step of transferring said request includes the steps of:	OWS 2.0 is computer-implemented	Dr. Shamos did not prove that these References meet the preamble. In addition, my response to Dr. Shamos's section on Oracle Developments sets forth further reasons why these References do not anticipate any claims of the patents-in-suit.
16(b): transferring said request from said HTTP-compliant device to a dispatcher, and	"The HTTP Server is the network layer component of the Oracle WebServer. It listens for incoming HTTP requests, delivers static files and runs simple CGI programs and hands off everything else to the Web Request Broker (WRB), where a request is handled by a server extension." ORCL 000764.	As discussed in the section below, Dr. Shamos failed to establish that these References have a "dispatcher" because these References do not disclose "dispatching" under either Oracle's or epicRealm's proposed construction.
16(c): dispatching said request to said page server	The WRB Dispatcher makes an informed selection of which page server (WRBX) should process the request based on dynamic information: which WRBX is free and which one is configured to run the desired service. Load balancing is also performed: "WRB Executable Engine (WRBX). One of a pool of processes that the WRB maintains continuously, so that HTTP requests requiring the execution of programs are not slowed down by the performance cost of spawning a new process. WRBX's are associated with WRB Cartridges and are created and destroyed according to workload." p. A-13. "The WRB dispatcher must decide what type of object is being requested. To do this, it examines the WRB configuration file, which maps virtual directories to WRB Services. ... The WRB Dispatcher performs dynamic load balancing between multiple	These References do not disclose "dispatching" under epicRealm's or Oracle's construction. EpicRealm's proposed construction for "dispatching" is "examining a request to make an informed selection of which page server should process the request based on dynamic information maintained about page servers, the dynamic information indicating which page server can more efficiently process the request, and sending the request to the selected page server." These References

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
	instances of a WRB Service, and the webmaster can configure each WRB Service to have a minimum and maximum number of instances. An instance corresponds to a process in a UNIX environment." Technical Note, p. 5	do not disclose maintaining dynamic information indicating which page server can more efficiently process the request nor do these References disclose making an informed selection of which page server should process the request. The dynamic information maintained by the WRB dispatcher only indicates which WRBXs are executing requests and which are free to process other requests. A busy WRBX cannot process a request. The dispatcher does not maintain information about which one of the free WRBXs can more efficiently process a request. Thus, these References do not disclose maintaining dynamic information indicating which page server can more efficiently process the request. Dr. Shamos argued that the maintained dynamic information is "which WRBX is free and which one is configured to run the desired service." As discussed above, information about which WRBX is free is not information about which page server can more efficiently process a request because a busy WRBX cannot process a request. Further, information about which WRBX is configured to run a desired service is not dynamic information because the configuration information is stored in a static configuration file. Similarly, the information about which WRBX is configured to run a desired service is not information about which

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
		page server can more efficiently process a request because it does not differentiate the between the relative efficiencies of page servers. A WRBX that is not configured to run a desired service is not an available page server to process a request, thus it is not considered in determining whether a page server can more efficiently process a request. In addition, these References do not disclose making an informed selection of which page server should process a request based on that dynamic information. It is my understanding that the WRB Dispatcher sends a request to a free WRBX to process the request. The WRB Dispatcher's selection of a "page server" is not informed because it does not distinguish between the free page servers that can process a request. Similarly, Dr. Shamos also failed to demonstrate that these References disclose "dispatching" under Oracle's construction. At a minimum, Dr. Shamos did not distinguish his analysis under epicRealm's and Oracle's proposed construction for "dispatching." Therefore, for the reasons set forth above, these References do not disclose "dispatching." In addition, Oracle's construction requires that the "dispatcher" make an "informed selection of which page server should process the request…." In these References, there is not an "informed

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)**	**Dr. Finkel's Analysis**
		selection" for the reasons set forth above. Thus, these References do not disclose "dispatching" under Oracle's proposed construction. Dr. Shamos contended that these references disclose "load balancing" but fails to explain what relevance the "load balancing" has to the claim terms. Indeed, there are many forms of load balancing that do not meet the limitations of the claims, including the form of "load balancing" disclosed by these References for the reasons set forth above.
17: The computer-implemented method in claim 15 wherein said step of processing said request includes the step of identifying said one or more data sources from which to retrieve said data	See '554 1b6 (configuration file). "[PL/SQL Agent] also needs to have access to all the details required to actually establish a permanent connection: typically a set of environment variables (ORACLE_HOME and ORACLE_SID for UNIX systems) and perhaps a SQL*Net connect descriptor. All of this information is referred to as a DCD (Database Connection Descriptor), and all of the DCDs known t o the PL/SQL Agent are stored in a configuration file named owa.cfg (DCDs where known as Web Agent Services in Webserver 1.0). . . . DCDs are defined in the PL/SQL Agent's configuration file, but they are activated by creating a directory mapping in the Web Request Broker's configuration file." ORCL000772.	Not asserted.
18: The computer-implemented method in claim 15 wherein said step of dynamically generating said Web page includes the step of dynamically retrieving	Oracle WebServer User's Guide Release 2.0.2 Production, at page ORCL000833 discloses a web page that contains data retrieved in response to user's request. In particular, column headings and data in the table has been retrieved from a database table. "LiveHTML code is formatted as SGML comments, so that it is ignored should the file ever find its way t o the browser unparsed. The format for LiveHTML codes, therefore, is the following: <!--#command	Not asserted.

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)**	**Dr. Finkel's Analysis**
said data from said one or more data sources	tag1="value1" tag2="value2" --> The tags are arguments to the commands, most of which actually only accept one of the possible tags. The possible commands and their associated tags are as follows: . . . include. This command specifies that a file is t o be included in the generated HTML page at this point." ORCL000778-779.	
19: The computer-implemented method in claim 15 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more data sources	"To do this, the Web Listener passes requests intended for these programs to the WRB Dispatcher, which maintains a pool of processes to which it can assign the requests." p. 1-7. This "pool of processes" is a connection cache	Not asserted.
20: The computer-implemented method in claim 15 wherein said step of processing said request includes the step of logging into said one or more data sources	"A URL that invokes the PL/SQL Agent must specify a DCD (Database Connection Descriptor). This is an OS file maintained by the WebServer that provides the username, password, database, and other information to be used to establish the database connection." p. 1-14. "At the lowest level, Oracle; internal API allows us t o establish a connection to an Oracle7 Server using SQL*Net in one step, and actually logging on to the database in a separate step." Technical Note, p. 11	Not asserted.
21: The computer-implemented method in claim 15 wherein said step of dynamically generating said page includes the step of maintaining a page cache containing said page	"Page caching. Ordinarily, when the Web Listener opens a requested file, the file remains open and mapped into memory until all clients using the file are finished with it, at which point the Web Listener closes the file and frees the memory mapping associated with it. The Web Listener allows you to specify files to be cached. Cached files are opened when a client requests them, and remain open for the life of the Web Listener process. This optimizes access times for files, such as your home page, that are requested often." p.	Not asserted.

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)**	**Dr. Finkel's Analysis**
	1-5	
22: The computer-implemented method in claim 15 wherein said page server includes tag-based text templates for configuring said page	LiveHTML comprises tag-based text templates. See '554 7	Not asserted.
23: The computer-implemented method in claim 22 wherein said step of processing said request further includes the step of inserting said dynamically retrieved data from said one or more data sources into said tag-based text templates	See '554 8.	Not asserted.
24: The computer-implemented method in claim 22 wherein at least one of said tag-based text templates drives a format of the data dynamically retrieved from said one or more data sources in response to said request	!@@Section 5 (e.g., page 5-5, MAC003147) describes the use of tag-based text templates in PL/SQL procedures to drive the format of the data in creating the generated page. "The Oracle WebServer Developer's Toolkit is a set of procedures that help the customer create programs to generate dynamic documents that are easily presented in HTML." p. 1-2 (MAC003052). HTML determines the format of a web page	Not asserted.
25: The computer-implemented method in claim 22 wherein said tag-based text templates include HTML templates	See '554 8.	Not asserted.

Oracle WebServer User's Guide Release 2.0 Production, Part No. A23646-1 and Oracle WebServer 2.0 Technical Note

'335 Claim Language	**Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)**	**Dr. Finkel's Analysis**
26: The computer-implemented method in claim 15 wherein said step of processing said request further includes the step of dynamically updating data at said one or more data sources	"You can also use a cursor to update a table's contents. To do this, use a FOR UPDATE clause to lock the rows in the table. See "Using FOR UPDATE" in the PL/SQL User's Guide and Reference for more information. Sometimes, you may need to use cursor variables. which are not associated with a query until runtime. This is a form of dynamic SQL." p. B-19. UPDATE is similar to SELECT in that it takes a predicate and operates on all rows that make the predicate TRUE. For example: UPDATE Customers SET ADDRESS = null WHERE LNAME = ' Subciak ' ORCL000972.	Not asserted.
27: The computer-implemented method in claim 15 wherein said step of processing said request further includes the step of processing an object handling extension	"The Java Interpreter includes classes for native Oracle7 access via PL/SQL and dynamic creation of HTML documents. For customers who wish to employ their own programming language of choice, Oracle WebServer 2.0 offers an easy-to-use API allowing you to build your own server-side plugins ranging from existing legacy applications written COBOL to modern CORBA or COM/OLE based object-oriented systems written in for instance C++ or Smalltalk." ORCL000762-763. COM and OLE are object-handling extensions, as I understand that term to be used in the Patents	Not asserted.
28: The computer-implemented method in claim 27 wherein said object handling extension is an OLE extension	See '335 13. OLE is taught expressly	Not asserted.
29(a): A computer-implemented method comprising the steps of:	OWS 2.0 is computer-implemented	Not asserted.
29(b): transferring a request from an HTTP-	"The HTTP Server is the network layer component of the Oracle WebServer. It listens for incoming HTTP requests, delivers static	Not asserted.

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
compliant device to a dispatcher	files and runs simple CGI programs and hands off everything else to the Web Request Broker (WRB), where a request is handled by a server extension." ORCL 000764.	
29(c): maintaining dynamic information regarding data sources a given page server may access	"The WRB dispatcher must decide what type of object is being requested. To do this, it examines the WRB configuration file, which maps virtual directories to WRB Services. For instance, the Web Request Broker may be configured to dispatch all URL requests beginning with /java to the integrated Java Interpreter. However, if no suitable WRB Service is found, the request is passed back to the Web Listener and standard processing continues. The WRB Dispatcher performs dynamic load balancing between multiple instances of a WRB Service, and the webmaster can configure each WRB Service to have a minimum and maximum number of instances. An instance corresponds to a process in a UNIX environment." ORCL000766. "The Dispatcher continually adjusts the load by controlling how many WRBXs are running at a given time, subject to certain parameters. These parameters are set by the Webserver Administrator, who sets the maximum and minimum number of WRBXs running for each WRB Sei-vice. The Dispatcher creates new WRBXs as needed and connects them to the appropriate WRB Services. The Dispatcher also keeps track of which WRBXs are executing requests and which are free." ORCL000806.	Not asserted.
29(d)(1): dispatching said request to an appropriate page server based on said request and based on said dynamic information	The WRB Dispatcher makes an informed selection of which page server (WRBX) should process the request based on dynamic information: which WRBX is free and which one is configured to run the desired service. Load balancing is also performed: "WRB Executable Engine (WRBX). One of a pool of processes that the WRB maintains continuously, so that HTTP requests requiring the execution of programs are not slowed down by the performance cost of spawning a new process. WRBX's are associated with WRB	Not asserted.

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
	Cartridges and are created and destroyed according to workload." p. A-13. "The WRB dispatcher must decide what type of object is being requested. To do this, it examines the WRB configuration file, which maps virtual directories to WRB Services. ... The WRB Dispatcher performs dynamic load balancing between multiple instances of a WRB Service, and the webmaster can configure each WRB Service to have a minimum and maximum number of instances. An instance corresponds to a process in a UNIX environment." Technical Note, p. 5	
29(d)(2): said page server receiving said request and	The request must be received to be processed. "The PL/SQL Agent performs two major tasks which enable a stored procedure to generate a dynamic HTML page. The first is to launch the procedure and pass in any parameters that were part of the HTTP request." ORCL000771.	Not asserted.
29(d)(3): releasing said HTTP-compliant device to process other requests	"AII inter-process communication is handled by the transport independent WRB protocol. Webserver 2.0 supports standard IPC mechanisms" Technical Note, p. 5. A standard IPC mechanism is Socket-Bind-Listen, hence SBL releasing is performed according to EpicRealm's contention. The Web Request Broker API document has a diagram at page ORCL01606867 that shows that WEB Application (such as PL/SQL Agent) sends REQ COMPLETE Response message, which is an act that is separate from merely receiving the request. HTTP Server receives the REQ COMPLETE Response message and frees up certain resources that would either allow processing of other requests or facilitate more efficient processing of already received requests. HTTP Server must free such resources because otherwise the REQ COMPLETE Response message has no purpose. ORCL01606867. Under the SBL theory, the INIT Response message sent from Web Application (such as PL/SQL Agent) to HTTP Server is an action, which is separate from merely receiving the request that informs the HTTP Server that it is	Not asserted.

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
	now able to offload the processing of the request to the now initialized WEB Application, prospectively freeing up the HTTP Server to process other requests once a request capable of being processed by the WEB Application is received.	
29(e): processing said request, said processing being performed by said page server while said HTTP-compliant device concurrently processes said other requests, and	Concurrency is taught expressly: "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), a program to be accessed through the CGI interface, or whether access to the file system of the machine on which the Listener resides is sufficient. If WRB access is required, the Listener passes the request to WRB Dispatcher for processing: then it returns to the task of listening for more incoming HTTP requests." p. 1-2. "Oracle's HTTP Server is designed to be a very scalable, high-performance network daemon. It is implemented as a multi-threaded, single-process asynchronous engine, allowing multiple concurrent requests to be processed at the same time using standard HTTP or HTTP over SSL" Technical Note, p. 3	Not asserted.
29(f)(1): dynamically generating a Web page in response to said request	"As documented in the CGI 1.1 specification, the Oracle Web Listener takes the contents of standard output and returns it to the Web browser that requested the dynamic HTML document." p. 6-6. "QUERY_FORM prints an HTML page with all the columns for the specified table. Invoke the procedure from a Web Browser with a URL like: http://yourhost:port_num/service_name/owa/query_form?the_table=emp . . . Invoking this procedure brings up a page that looks like [page shown at ORCL01816677]. Also see ORCL01816678-79.	Not asserted.
29(f)(2): said page including data dynamically retrieved from one or more data sources	"The Oracle WebServer ... draws on information from the database and the operating system's (0S) file system as necessary to respond to the request. The file system can be used for static (hardcoded) Web pages, or form scripts that do not access the database, and the database is used for Web	Not asserted.

'335 Claim Language	Dr. Shamos's Contentions (Quoted Directly from His Claim Charts)	Dr. Finkel's Analysis
	pages that are generated at runtime using "live" data." p. 1-2. Also: "LiveHTML code is formatted as SGML comments, so that it is ignored should the file ever find its way t o the browser unparsed. The format for LiveHTML codes, therefore, is the following: <!-- #command tag1="value1" tag2="value2" --> The tags are arguments t o the commands, most of which actually only accept one of the possible tags. The possible commands and their associated tags are as follows: . . . include. This command specifies that a file is t o be included in the generated HTML page at this point." ORCL000778-779.	