

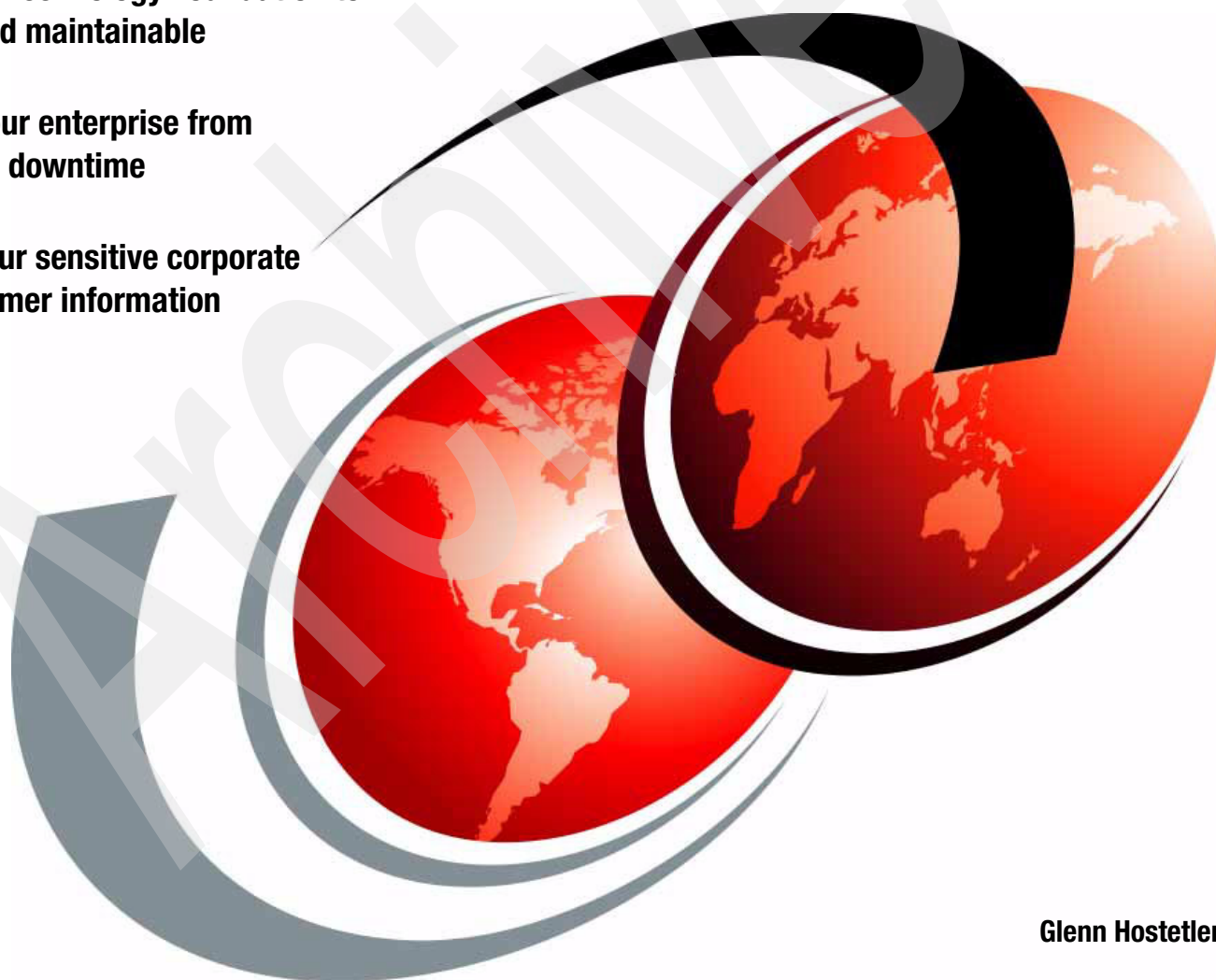
# IBM *e*server and JD Edwards EnterpriseOne Technology Foundation

## Ensuring a High Quality of Service

Configure Technology Foundation to  
be fast and maintainable

Protect your enterprise from  
expensive downtime

Secure your sensitive corporate  
and customer information



Glenn Hostetler

# Redbooks





International Technical Support Organization

**IBM @server and JD Edwards EnterpriseOne  
Technology Foundation:  
Ensuring a High Quality of Service**

**August 2005**

Archived

**Note:** Before using this information and the product it supports, read the information in “Notices” on page xi.

## **Second Edition (August 2005)**

This edition applies to IBM @server iSeries, IBM @server pSeries, IBM @server zSeries, and IBM @server xSeries, Domino, WebSphere, WebSphere Portal, Lotus Instant Messaging and Web Conferencing (Sametime), Lotus Team Workplace (QuickPlace), Network Dispatcher, Edge Server, IBM HTTP Server, DB2 Universal Database, IBM Directory Server, AIX, OS/400, Web Traffic Express, WebSEAL, Rational Robot, WebSphere MQ (IBM); Technology Foundation, EnterpriseOne (Oracle); Windows, Active Directory (Microsoft); Java, Java System Directory Server (Sun); Presentation Server (Citrix); Intel Processors (Intel); IBM HTTP Server (powered by Apache); Cisco SSL Accelerators (Cisco); LoadRunner (by Mercury Interactive); and SAP Version.

© Copyright International Business Machines Corporation 2005. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

# Contents

<b>Figures</b> .....	vii
<b>Tables</b> .....	ix
<b>Notices</b> .....	xi
Trademarks .....	xii
<b>Preface</b> .....	xiii
The team that wrote this redbook. ....	xiii
Become a published author .....	xiv
Comments welcome. ....	xv
<b>Chapter 1. What Technology Foundation is</b> .....	1
<b>Part 1. Before installation</b> .....	5
<b>Chapter 2. Methods and requirements</b> .....	7
2.1 Methodology for producing a physical architecture. ....	8
2.2 A logical architecture. ....	10
<b>Chapter 3. Logical architecture selection</b> .....	13
3.1 Key logical architecture issues for Technology Foundation .....	14
3.1.1 Fault tolerance .....	14
3.1.2 Degree of security. ....	19
3.2 Supported, standard logical architectures. ....	23
3.2.1 Standard Security, Highly Available .....	24
3.2.2 Standard Security, Continuously Available .....	25
3.2.3 Highly Secure, Highly Available .....	26
3.2.4 Test architecture .....	28
3.3 Allowed customizations to logical architectures .....	29
3.3.1 Connecting to third-party external directory servers (LDAPs). ....	29
3.3.2 Downgrading availability for non-mission critical elements. ....	31
3.3.3 Increasing performance for remote locations .....	32
3.3.4 Very high internal security. ....	34
3.3.5 Providing Internet access .....	36
3.3.6 Secure Sockets Layer accelerators .....	38
3.3.7 Third-party IP sprayers .....	38
3.3.8 Third-party database vendors .....	38
3.4 Selecting appropriate fault tolerance and security .....	39
3.4.1 Determining fault tolerance requirements. ....	39
3.4.2 Determining security requirements .....	41
<b>Part 2. After installation</b> .....	45
<b>Chapter 4. Configuring and tuning Technology Foundation for high transaction volumes</b> .....	47
4.1 pSeries benchmark .....	48
4.1.1 Physical architecture. ....	49
4.1.2 Detailed logical architecture .....	51
4.2 Recommended parameter values and reasoning .....	52

4.2.1	Tuning parameters that prevent dropped transactions at high volumes . . . . .	53
4.2.2	Parameters that enhance response time . . . . .	59
4.2.3	Tempting but insignificant parameters . . . . .	63
4.3	xSeries . . . . .	65
4.4	iSeries . . . . .	66
4.5	Miscellaneous tuning tips . . . . .	66
4.5.1	Database configuration changes . . . . .	66
4.5.2	Watching for contention on the next numbers table . . . . .	66
4.5.3	Ensuring your test scripts do not lock tables. . . . .	67
4.5.4	Paying attention to how your test tool classifies dropped transactions. . . . .	67
<b>Chapter 5.</b>	<b>Managing Technology Foundation . . . . .</b>	<b>69</b>
5.1	Installation and maintenance roles . . . . .	70
5.2	Operations monitoring and management tools . . . . .	71
5.2.1	WebSEAL . . . . .	71
5.2.2	Network Dispatcher. . . . .	71
5.2.3	HTTP. . . . .	72
5.2.4	WebSphere Application Server. . . . .	73
5.2.5	Portal. . . . .	74
5.2.6	EnterpriseOne Web Server (JAS). . . . .	74
5.2.7	DB2 Universal Database. . . . .	75
5.2.8	System monitoring . . . . .	76
5.3	Maintenance . . . . .	76
5.3.1	Quality assurance environments for new software releases. . . . .	76
5.4	Commonly overlooked best practices for maintenance. . . . .	83
5.5	Beyond Technology Foundation . . . . .	84
5.5.1	Enhancing Technology Foundation with portlets . . . . .	84
5.5.2	Portal's Credential Vault . . . . .	85
5.5.3	Publicly accessible enterprises . . . . .	85
5.5.4	Running additional applications on the hardened J2EE application server . . . . .	86
5.5.5	Extending the architecture using Java Message Service . . . . .	86
5.5.6	Using WebSphere Enterprise Edition features to do business process modeling . . . . .	87
<b>Appendix A.</b>	<b>Sample logical architecture selection document. . . . .</b>	<b>89</b>
	Table of contents . . . . .	91
	Executive summary . . . . .	92
	Business overview . . . . .	92
	Business objectives . . . . .	93
	Acme's current environment . . . . .	94
	Current physical architecture . . . . .	94
	Current user community . . . . .	95
	Current security mechanisms . . . . .	95
	Recommended logical architectures . . . . .	96
	Initial rollout recommendation. . . . .	96
	Medium-term recommendation. . . . .	99
	Long-term recommendation . . . . .	100
	Rationale for architecture proposal. . . . .	101
	Required skills . . . . .	102
	Next steps . . . . .	105
	Sample proposal glossary . . . . .	106
<b>Glossary</b>	<b>. . . . .</b>	<b>109</b>
<b>Related publications</b>	<b>. . . . .</b>	<b>113</b>

IBM Redbooks .....	113
References .....	113
Online resources .....	114
How to get IBM Redbooks .....	114
Help from IBM .....	114
<b>Index</b> .....	<b>115</b>

Archived



# Figures

1-1	Overview of Technology Foundation layers . . . . .	2
2-1	Organizations, process steps, and work products . . . . .	9
2-2	A sample logical architecture . . . . .	11
2-3	Logical architecture to physical machine sample mappings . . . . .	12
3-1	Distributed workload . . . . .	14
3-2	Nondistributed workload . . . . .	15
3-3	High availability mode works . . . . .	16
3-4	One machine becomes unavailable . . . . .	16
3-5	System becomes unavailable . . . . .	17
3-6	One of three machines becomes unavailable . . . . .	17
3-7	Continuously available workload during planned upgrade . . . . .	18
3-8	Logical architecture with standard security: Load balancers and HTTP in DMZ . . . . .	20
3-9	Highly secure logical architecture: Reverse proxy in DMZ . . . . .	22
3-10	Standard Security, High Availability logical architecture . . . . .	24
3-11	Standard Security, Continuously Available logical architecture . . . . .	25
3-12	Highly Secure, Highly Available logical architecture . . . . .	26
3-13	Highly Secure, Continuously Available logical architecture . . . . .	27
3-14	Test architecture for pilot projects . . . . .	28
3-15	Logical architecture variation: External corporate directory server . . . . .	30
3-16	Logical architecture variation: No redundancy in non-critical elements . . . . .	31
3-17	Logical architecture variation: Downgraded availability for non-critical components . . . . .	32
3-18	Logical architecture variation: Increasing performance with Edge Servers . . . . .	33
3-19	Logical architecture variation: Very highly secure . . . . .	35
3-20	Logical architecture variation: Very highly secure with Internet-ready performance . . . . .	37
4-1	Physical architecture for benchmark . . . . .	49
4-2	Physical architecture for horizontal scaling portion of benchmark . . . . .	50
4-3	Detailed logical architecture . . . . .	51
5-1	Technology Foundation double-layered symbols . . . . .	77
5-2	Technology Foundation triple-layered symbols . . . . .	78
5-3	Technology Foundation symbol conventions . . . . .	78
5-4	Technology Foundation infrastructure . . . . .	79
5-5	Development and Unit Test Layer . . . . .	80
5-6	Integration Layer . . . . .	81
5-7	Test Layer simple development environment . . . . .	82
5-8	Test Layer complex development environment . . . . .	82
5-9	Production Layer . . . . .	83
A-1	Acme's physical architecture today . . . . .	94
A-2	Recommended logical architecture for Acme . . . . .	97
A-3	Software component and hardware platform map . . . . .	98
A-4	Recommended medium-term logical architecture for Acme . . . . .	99
A-5	Recommended long-term logical architecture for Acme . . . . .	101

Archived

# Tables

1-1	Contents of Technology Foundation Version 4 and 5 . . . . .	3
2-1	Steps to produce a physical architecture . . . . .	8
3-1	Degraded and non-degraded sets . . . . .	15
3-2	Supported logical architecture options . . . . .	23
3-3	Levels of security . . . . .	42
4-1	Hardware for HTTP, WebSphere, and EnterpriseOne servers . . . . .	50
4-2	Database servers . . . . .	51
5-1	Skill sets to maintain Technology Foundation . . . . .	70
5-2	Configuration requirements . . . . .	81
A-1	Required skills to install and maintain the proposed architecture . . . . .	103
A-2	Potential action items . . . . .	105

Archived

# Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.*

*The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law.* INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.


This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

## COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

## Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

@server®	DB2 Universal Database™	Redbooks™
iSeries™	DB2®	RS/6000®
pSeries®	Enterprise Storage Server®	Sametime®
xSeries®	IBM®	Tivoli Enterprise™
AIX®	Lotus®	Tivoli Enterprise Console®
AS/400®	Netfinity®	Tivoli®
ClearCase®	OS/400®	WebSphere®
ClearQuest®	QuickPlace®	Workplace™
Domino®	Redbooks (logo)  ™	

The following terms are trademarks of other companies:

Enterprise JavaBeans, EJB, Java, Javadoc, JavaBeans, JavaServer, JavaServer Pages, JDBC, JSP, JVM, J2EE, Sun, Sun Microsystems, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft Internet Explorer, Microsoft, Windows NT, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Pentium, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

# Preface

Technology Foundation from Oracle is an integration of the IBM® world-class middleware with JD Edwards EnterpriseOne application software. Technology Foundation provides a robust Web interface complete with collaboration and portal technology. This IBM Redbook covers the Technology Foundation components. These include IBM WebSphere® Application Server, WebSphere Portal, Lotus® Collaboration (IBM Lotus Team Workplace™ (QuickPlace®), IBM Lotus Instant Messaging and Web Conferencing (Sametime®), and Lotus Domino®), IBM HTTP Server, WebSphere Edge Server's Network Dispatcher, and DB2® Universal Database™ (UDB).

Part one discusses the steps to help you prepare for installing Technology Foundation. It helps you to determine failover and security requirements prior to ordering hardware. Plus, it explains how to select a supported architecture for JD Edwards' EnterpriseOne Technology Foundation that:

- ▶ Protects your enterprise from expensive downtime
- ▶ Secures your sensitive corporate and customer information
- ▶ Is properly configured, fast, and maintainable

Part two, which can mean the difference between the success and failure of a project, examines issues that arise after you install Technology Foundation. It provides guidance to help you configure and tune Technology Foundation for high transaction volumes. This part also provides best practices to help you manage and maintain Technology Foundation. It is essential to properly define the system's logical architecture. No amount of tuning can fix fundamental architecture problems.

This IBM Redbook is written for clients who want to improve the availability of their business by implementing a proven solution with IBM middleware and JD Edwards application software. It also targets Oracle clients who are preparing to purchase EnterpriseOne Technology Foundation. It helps to make important architectural choices that will determine the limits of performance, security, and failover. And this redbook is designed to help hardware vendors who work with the JD Edwards Technology Foundation clients to help them with their choices. Appendix A, "Sample logical architecture selection document" on page 89, is provided specifically for such hardware vendors as an example of how to properly assemble a logical architecture proposal for their clients.

## The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working for the International Technical Support Organization (ITSO), Rochester Center.

**Glenn Hostetler** is an Enterprise Architect in the IBM Business Partner Technical Services (BPTS) branch. He enables independent software vendors (ISVs) to properly use and deploy WebSphere-related technologies. Applying over 20 years of experience in software developments, he works with ISV resellers to architect, design, develop, and field Java™ 2 Platform, Enterprise Edition (J2EE™)-based information systems that benefit the unique needs of each business. Over the past eight years, he has architected and implemented numerous large, widely distributed systems that provide core, day-to-day business operations for Fortune 500 corporations. His role in BPTS has allowed him to specialize in architecting solutions that address performance, scaling, high availability, and security.

Thanks to the following people for their contributions to this project:

Mark Giles  
Bob Hieronymus  
Jagan Karuturi  
Douglas McGarrie  
Will Wang  
IBM Austin, Texas

Boyd Fenton  
Larry Grubbs  
Sandor Hasznos  
Rob Jump  
Barry Watts  
IBM Denver, Colorado

Eric Gong  
IBM Fremont, California

Leslie Clark  
Clark Scholten  
IBM Rochester, Minnesota

Susan Powers  
ITSO, Rochester Center

## Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners, and/or Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

[ibm.com/redbooks/residencies.html](http://ibm.com/redbooks/residencies.html)



## Comments welcome

Your comments are important to us!

We want our Redbooks™ to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- ▶ Use the online **Contact us** review redbook form found at:

[ibm.com/redbooks](http://ibm.com/redbooks)

- ▶ Send your comments in an Internet note to:

[redbook@us.ibm.com](mailto:redbook@us.ibm.com)

- ▶ Mail your comments to:

IBM Corporation, International Technical Support Organization  
Dept. JLU Building 107-2  
3605 Highway 52N  
Rochester, Minnesota 55901-7829

Archived

# What Technology Foundation is

Technology Foundation (informally referred to as “Tech Foundation”) is a collection of IBM software components for viewing JD Edwards’s EnterpriseOne software suite from a Web browser. Technology Foundation consists of several components. These components include WebSphere Application Server, WebSphere Portal, Lotus Collaboration (IBM Lotus Team Workplace (QuickPlace), IBM Lotus Instant Messaging and Web Conferencing (Sametime), and Lotus Domino), IBM HTTP Server, WebSphere Edge Server’s Network Dispatcher, and DB2 Universal Database (UDB).

**Note:** Network Dispatcher is part of Technology Foundation Version 5. It is not part of Version 4. Technology Foundation Version 4 includes WebSphere Version 4.x, while Technology Foundation Version 5 includes WebSphere Version 5. Network Dispatcher comes bundled with WebSphere Version 5, but not Version 4.

DB2 is integrated in OS/400® and, therefore, is included automatically for the IBM @server iSeries™ solution.

The components run on various hardware platforms. The platforms of choice are IBM @server xSeries® (Microsoft® Windows®-based), IBM @server pSeries® (AIX® UNIX®-based), and iSeries (OS/400 based) hardware. They are the preferred platforms for the Technology Foundation infrastructure because the software components are developed and most thoroughly tested on these platforms.

Technology Foundation greatly reduces system administration costs. Without Technology Foundation, EnterpriseOne can only be accessed via fat client applications. With Technology Foundation, EnterpriseOne can be served from a Web site at a central data center.

**Note:** The EnterpriseOne fat client can also be served indirectly through presentation servers such as those offered by Citrix or Microsoft. However, software suffers from the drawbacks of attempting to extract a thin client interface from an application originally designed as a thick client. Such drawbacks become increasingly obvious as the number of users increases. Presentation servers, especially in large enterprise environments, typically suffer a poor cost per user ratio.

System administrators no longer need to manage individual applications on thousands of corporate PCs and mobile computers. Upgrades are performed in minutes rather than days, and client software version conflicts are avoided.

To clarify the role of Technology Foundation, refer to Figure 1-1. It compares an EnterpriseOne deployment with and without Technology Foundation. It also provides a simplified overview of each layer and the approximate functions that are performed.

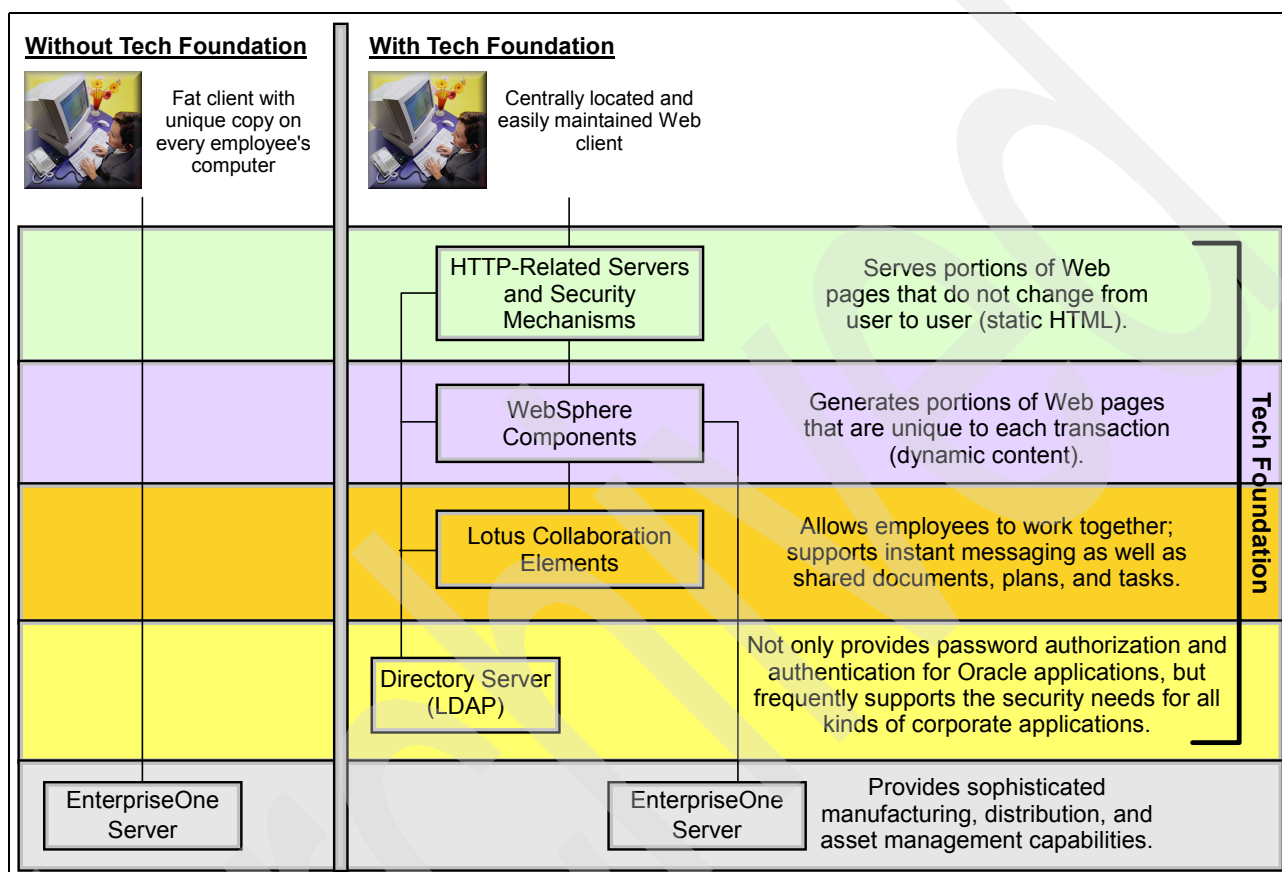


Figure 1-1 Overview of Technology Foundation layers

The Technology Foundation architectures represented in this IBM Redbook can be roughly stratified in this manner, although the layers are more complex than shown here. For example, the *HTTP-related Servers and Security Mechanisms* layer includes multiple servers and firewalls, as well as multiple software components such as WebSphere Edge Server's Network Dispatcher, IBM HTTP Server, and possibly WebSEAL. The *WebSphere Components* layer can include WebSphere Application Server, WebSphere Portal Server, Lotus Collaboration, and the EnterpriseOne Web Server. The *Lotus Collaboration Elements* layer can include Lotus Instant Messaging and Web Conferencing, Lotus Team Workplace, and Lotus Domino Server. Despite these simplifications, Technology Foundation can be accurately modeled with these four layers.

The scope of this redbook is limited to the issues surrounding the four layers that compose the Technology Foundation front end. The EnterpriseOne back-end software components are considered out of the scope of this document and, therefore, are not described. JD Edwards provides numerous other documents about establishing failover and security, tuning, and extending EnterpriseOne.

The licenses for the infrastructure Technology Foundation products are limited. They cannot be used for applications other than EnterpriseOne and its supporting infrastructure without an expanded license purchase. For example, the JD Edwards Collaborative Portal, which is a part of Technology Foundation, permits the use of portlets for JD Edwards EnterpriseOne, Lotus Instant Messaging and Web Conferencing, and Lotus Team Workplace. The use of the Collaborative Portal with portlets from other applications, such as SAP, requires the purchase of an unrestricted-use license of WebSphere Portal. This unrestricted version is available from IBM and entitles you to a discount for having purchased JD Edwards Technology Foundation.

Table 1-1 shows the contents of Technology Foundation Version 4 and 5.

*Table 1-1 Contents of Technology Foundation Version 4 and 5*

	<b>Technology Foundation Version 4</b>	<b>Technology Foundation Version 5</b>
Edge Server <sup>1</sup>	Included	Included
Network Dispatcher <sup>2</sup>	Not included	Included in WebSphere Edge Server, Version 5
Domino Enterprise (upgrade required to support failover of Lotus Instant Messaging and Web Conferencing and Lotus Team Workplace)	Not included	Not included
WebSEAL <sup>3</sup>	Not included, although not necessarily needed depending on logical architecture selection.	Not included, although not necessarily needed depending on logical architecture selection.
IBM Directory Server	Included	Included
WebSphere Application Server	Included	Included
WebSphere Portal	Included	Included
Collaborative Server	Included	Included
EnterpriseOne Web Server	Included	Included
IBM HTTP Server	Included	Included
DB2 UDB	Included	Included
<ol style="list-style-type: none"> <li>1. In Version 5, Edge Server is renamed to Caching Proxy Component (once referred to as Web Traffic Express). It comes as part of the WebSphere Application Server Edge Components offering (formerly known as WebSphere Performance Package). This redbook uses the Version 4 terminology of Edge Server.</li> <li>2. In Version 5, Network Dispatcher is renamed to Load Balancer. It comes as part of the WebSphere Application Server Edge Components offering. This redbook uses the Version 4 terminology of Network Dispatcher since the tuning efforts described in Chapter 4, "Configuring and tuning Technology Foundation for high transaction volumes" on page 47, use Version 4. Plus this avoids confusion with other products that perform the task of load balancing.</li> <li>3. Edge Server Version 5 comes with Network Dispatcher and a forward and reverse proxy cache called Web Traffic Express. For some customers, it may be possible to substitute Web Traffic Express for the WebSEAL reverse proxy. However, at the time of this writing, Web Traffic Express did not support authentication integration through many commonly used directory servers such as Microsoft Active Directory, Sun™ System Directory Server, or IBM Lotus Domino Server.</li> </ol>		

Table 1-1 is offered as a general overview of the contents of Technology Foundation. It is superseded by the licensing agreement.

In general, Technology Foundation Version 4 is defined by WebSphere and Portal Versions 4.x, while Technology Foundation Version 5 is defined by WebSphere and Portal Versions 5.x

The specific version information of each product that composes Technology Foundation is complex. Entire version ranges are supported for both Technology Foundation versions, and different hardware platforms and operating systems support different product ranges. In addition, the supported versions between components depend on the versions of other components. For example, an upgrade to the Portal server in Technology Foundation Version 4 can assume upgrades to the directory server or can require fix packs to the database.

The Technology Foundation licensing agreement is the best and final source to determine your desired product combinations and required upgrades.



# Part 1

## Before installation

This part discusses the steps that must occur before you can install Oracle's Technology Foundation. It describes the first step of a four-step process for producing a physical architecture for a JD Edwards EnterpriseOne Technology Foundation deployment. This physical architecture specifies what server hardware to buy, what software to install on them, how to interconnect the servers, and their cost. The four steps in this process include:

1. Logical architecture definition
2. Physical topology definition
3. Full specification for physical architecture
4. Proposal submission

The first step involves determining security and high availability requirements and using these requirements to select from a limited set of supported logical architectures. Then, the hardware vendor works with IBM sizing experts and the client to select a physical architecture from the finite set of supported physical topologies, thereby completing the second step. The third step, sizing the architecture to produce the final shopping list, is a proprietary IBM process and is not discussed in this redbook. The fourth step is performed by the hardware vendor. Based on the requirements for the number of CPUs, memory, disk space, etc., the vendor recommends a hardware configuration. Pricing information is added, and a formal proposal is presented to the end client that outlines the recommended physical architecture.

These steps are described in more detail in 2.1, "Methodology for producing a physical architecture" on page 8. The steps typically take place during the *sizing process* that accompanies the purchase of hardware. The client's selected hardware vendor offers assistance at each stage. The hardware vendor may also seek assistance from the IBM and Oracle International Competency Center (ICC).

This part focuses on the first step, logical architecture definition. Steps 2, 3, and 4 tend to depend on proprietary information, on information that varies from vendor to vendor, or on data that evolves too rapidly to document in this book. Consequently, these steps are discussed in this redbook at a summary level only.

Archived





## Methods and requirements

This chapter discusses the method that you must follow before you install Technology Foundation. Specifically, you must determine your failover and security requirements. These requirements help to determine the selection from the list of standard, supported logical architectures that are outlined in Chapter 3, “Logical architecture selection” on page 13. This chapter also defines what a logical architecture is, within the context of this redbook.

## 2.1 Methodology for producing a physical architecture

JD Edwards' methodology for producing physical architectures is a four-step process. Table 2-1 outlines each step. It includes the inputs required for the step, the output produced by the step, and the public or proprietary resources required for the execution of that step.

Table 2-1 Steps to produce a physical architecture

Step	Process step	Input	Output	Available resources
Step 1	Logical architecture definition	<ul style="list-style-type: none"> <li>▶ Feature set requirements</li> <li>▶ Fault tolerant requirements</li> <li>▶ Lightweight Directory Access Protocol (LDAP) requirements</li> <li>▶ Security requirements</li> </ul>	One of the six approved logical architectures defined in this document. The choice can be customized within the guidelines specified in 3.3, "Allowed customizations to logical architectures" on page 29.	<ul style="list-style-type: none"> <li>▶ Feature sets</li> <li>▶ Fault tolerant configurations</li> <li>▶ Security configurations</li> <li>▶ LDAP configurations</li> </ul>
Step 2	Physical topology definition	<ul style="list-style-type: none"> <li>▶ Logical topology from Step 1</li> <li>▶ Rough usage requirements</li> <li>▶ Rough cost requirements</li> <li>▶ Time to market requirements</li> <li>▶ Corporate standards</li> </ul>	An approved and supported physical topology, customized within guidelines. (A physical topology differs from a physical architecture in that it has not yet been sized.)	<ul style="list-style-type: none"> <li>▶ Usage model</li> <li>▶ Approved standard topologies</li> <li>▶ Schematics</li> <li>▶ Machine definitions</li> </ul>
Step 3	Full specification for physical architecture	<ul style="list-style-type: none"> <li>▶ Physical topology from Step 2</li> <li>▶ Sizing values (portal complexity, usage volumes, hours of operation, characteristics of peak load, cost restrictions, corporate standards for hardware, the client's technology roadmap, non-degraded failover requirements, etc.)</li> </ul>	The physical architecture consisting of: <ul style="list-style-type: none"> <li>▶ A part requirement list</li> <li>▶ A schematic</li> </ul>	Capacity model (proprietary)
Step 4	Proposal submission	Physical architecture from Step 3	Formal proposal with: <ul style="list-style-type: none"> <li>▶ Pricing</li> <li>▶ Available hardware options</li> </ul>	Pricing models

Step 1, logical architecture definition, defines what software components will exist on which generic physical hardware. In Step 2, these hardware *boxes* are precisely specified and are no longer general. Step 2 defines whether the boxes are pSeries running AIX UNIX, xSeries running Windows or Linux®, iSeries running OS/400, Integrated xSeries Servers hosted on an iSeries servers, logical partitions (LPARs) within one of these hardware platforms, or even partitions that span multiple machines (as blade servers can be configured).

The physical topology shows the kinds of hardware that will be used for the solution, but this information is not precisely sized. Step 3 refines the physical topology into a fully specified and precisely sized physical architecture requirements list. A network schematic shows how the elements in this requirement list fit together. Now that the usage patterns are defined, a recommended cell, node, and clustering topology accompanies the schematic. The amount

of necessary memory, number of CPUs, volume of disk drives, number of hardware machines, etc. is fully defined in the requirements list. However, only the hardware vendor can determine the size or brand of disk drives that they are currently selling or the increments in which memory is currently available. Consequently, during Step 4, only the IBM hardware vendor can assemble a formal proposal that uses the latest hardware configurations and latest pricing structures to meet the requirements of the physical architecture.

The specification defining exactly what individual or organization does what step is really a localized and rapidly evolving task, which is beyond the scope of this redbook. There is a reasonably complex work flow that engages multiple organizations. Each organization is involved at nearly every step. The organizations that are involved include:

- ▶ Oracle’s Global Advanced Technology Services (GATS)
 

Oracle’s services group that performs installations of EnterpriseOne-related products for clients
- ▶ Hardware vendors
 

The hardware vendors from which Oracle clients choose to purchase their IBM hardware
- ▶ IBM and Oracle International Competency Center (ICC)
 

An IBM organization that resides within Oracle to assist in the determination of the appropriate hardware given the needs of Oracle’s end users
- ▶ End client
 

Defines their requirements and preferences
- ▶ IBM Global Services staff assistance with the process is available, if appropriate
 

IBM Business Continuity Services staff may also assist, particularly if the client requires disaster recovery. See “Disaster recovery” on page 18 for more information.

Figure 2-1 attempts to show the generalized flow, while always in flux.

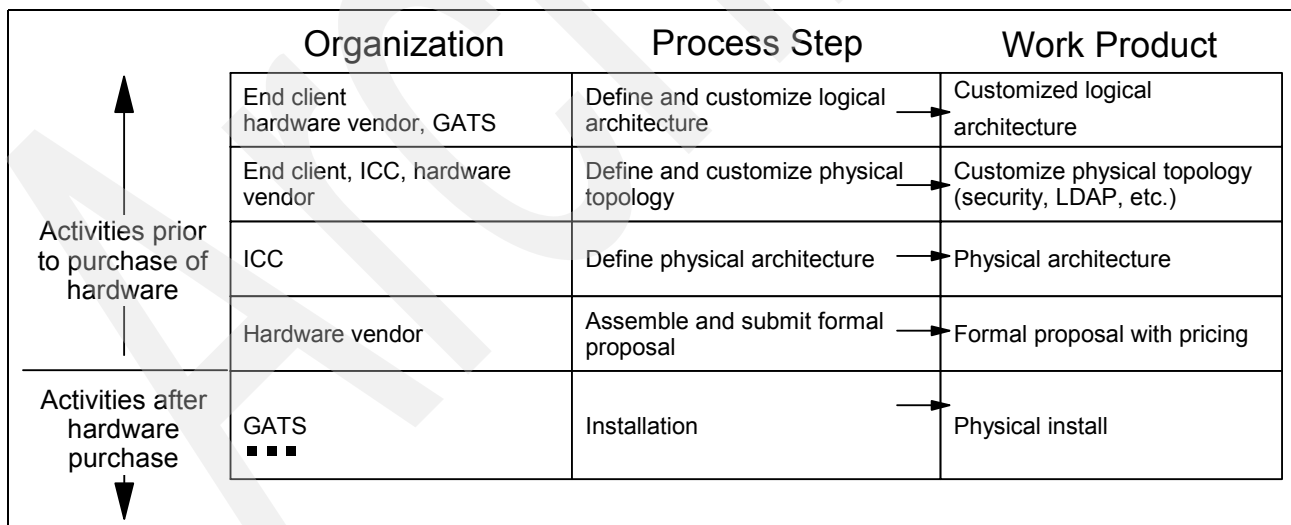


Figure 2-1 Organizations, process steps, and work products

## 2.2 A logical architecture

Architects usually have different notions of what constitutes a logical architecture. One opinion is that a logical architecture shows the interrelationships of a system's software components irrespective of hardware. Another common opinion is that logical architectures represent clusters of related pieces of software by business function (such as warehousing, inventory, or cash management). Still others group the logical architecture components by function (such as directory service software, Web servers, or firewalls).

The appropriate approach depends on the intended use and audience of the architecture. Since the logical architectures in this book are intended to be immediately translated into physical topologies for different platforms—either OS/400 iSeries, AIX pSeries, or Windows xSeries machines—the logical architecture elements here are grouped according to clusters of software components that are likely to end up on the same physical hardware.

**Note:** This approach can be understandably criticized since it introduces concepts of physical architecture—the allocation of software to specific boxes—into what is referred to in this redbook as a *logical architecture*. A classical approach was initially attempted. It added finely distinguished steps to the physical architecture definition process. These steps are perceived to add more overhead than value. The classical approach was discarded in favor of this simpler, concise, integrated approach.

Figure 2-2 shows a sample Technology Foundation architecture. The software groupings are represented by a surrounding dotted line. This shows that the group is likely to reside on the same physical machine or logical partition (in the case of iSeries or pSeries). Each group of software components can be mapped into one of many supported hardware platforms during the sizing process.

**Note:** The WebSphere database is composed of the Portal, Member Services, and WebSphere databases. It is represented here as one unit to simplify the diagram.

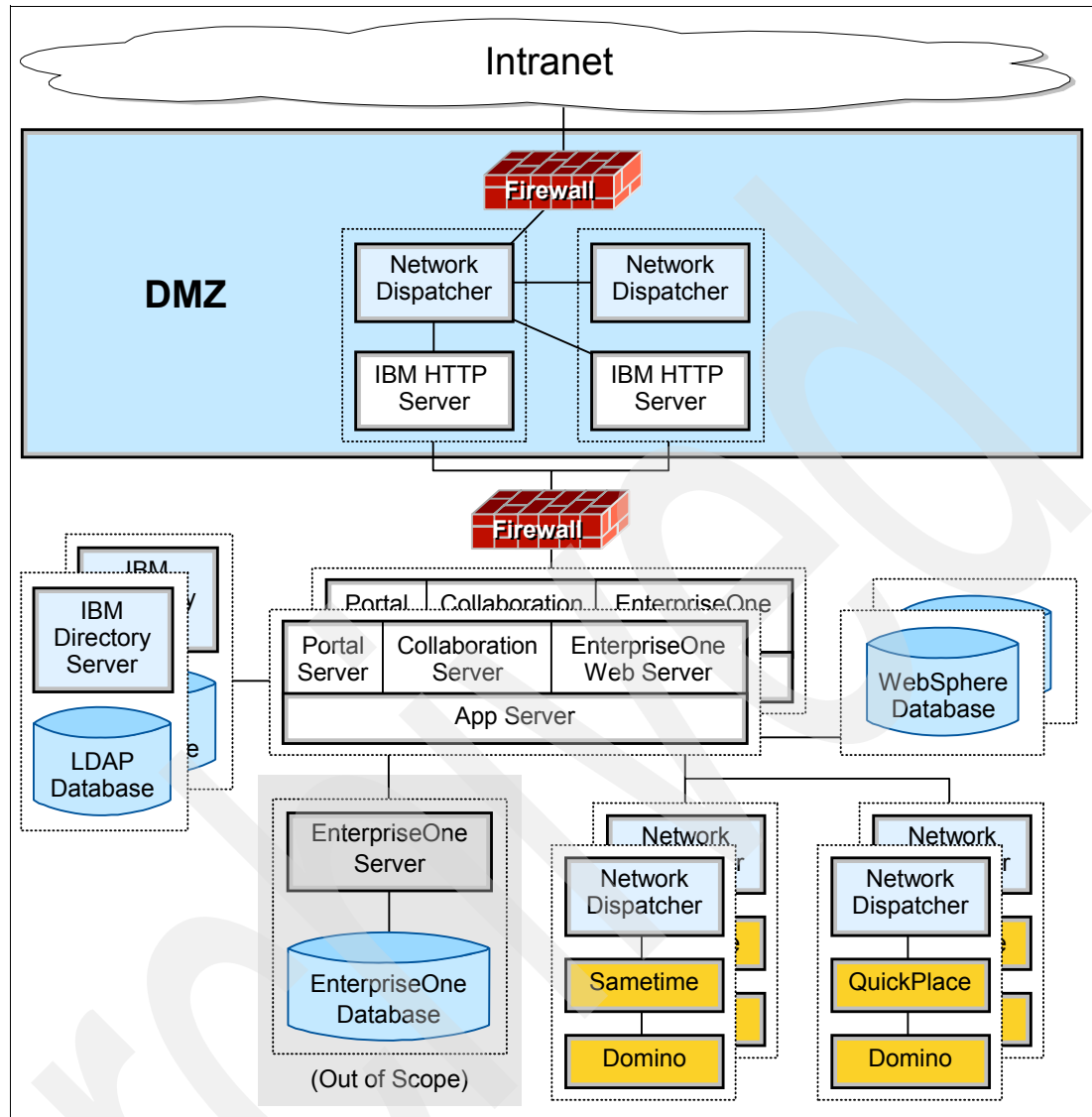


Figure 2-2 A sample logical architecture

Figure 2-3 shows some of the potential mappings. It illustrates how a single component grouping from the logical architectures represented in this redbook may map to an Intel® machine, to an LPAR on iSeries or pSeries hardware, to multiple horizontally-scaled machines, or to several other variations. Because the number of potential mappings is virtually infinite, JD Edwards appropriately limits the supported choices. By limiting the supported end physical architectures, JD Edwards allows for all supported options to be thoroughly tested and debugged, which improves the quality and performance of the solution for end users.

**Note:** The finite set of physical realizations are maintained internally by JD Edwards and evolve too rapidly to be covered in this redbook.

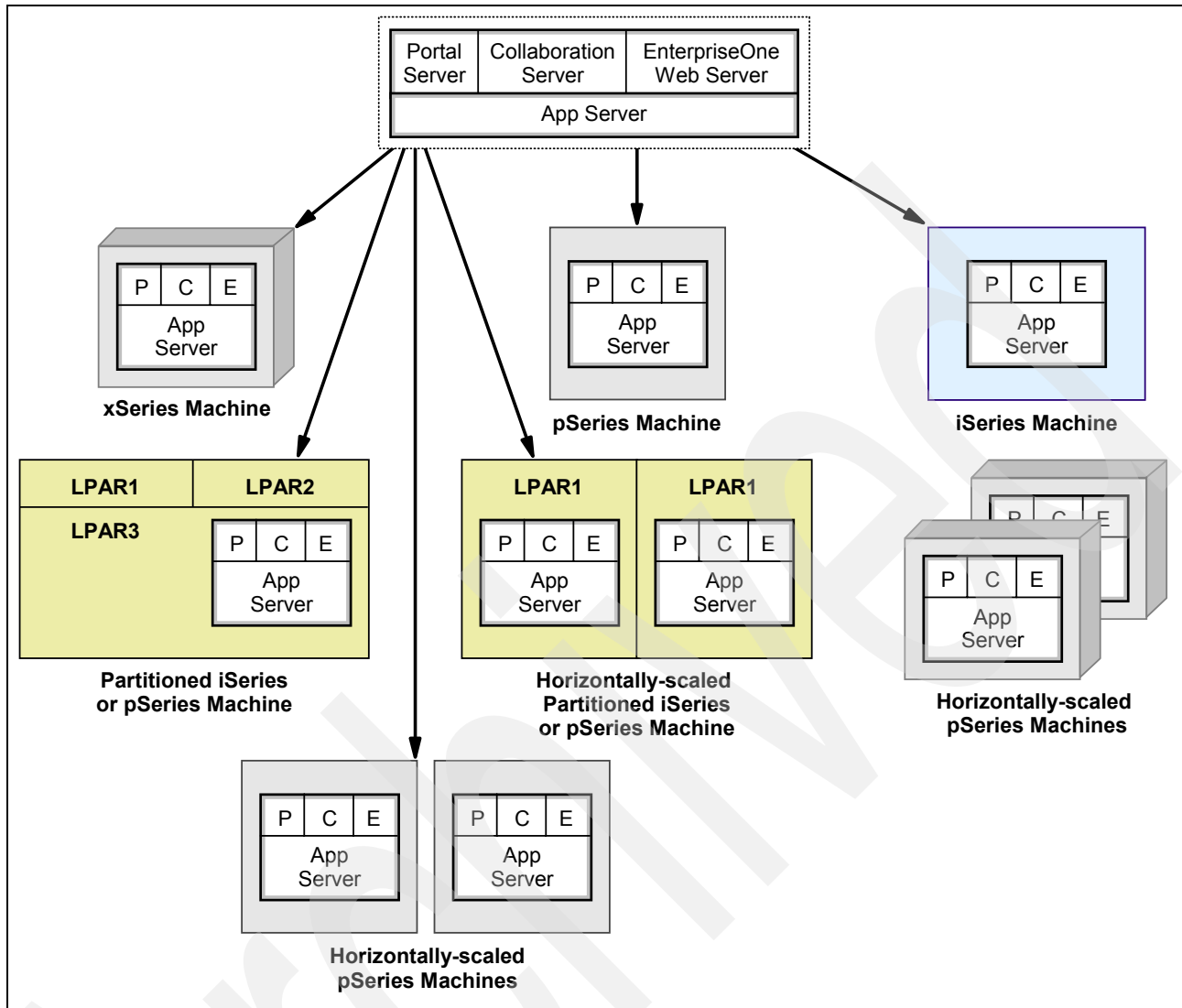


Figure 2-3 Logical architecture to physical machine sample mappings

## Logical architecture selection

This chapter discusses the issues that you face when selecting your Technology Foundation logical architecture. You must decide the appropriate degree of fault tolerance and security. Plus you must select a standard logical architecture and, if necessary, customize it. In doing so, you must work in conjunction with your hardware vendor to produce a physical architecture that meets both the immediate and growth needs of your business. You must resolve all fundamental architecture issues so that you know what to buy.

This chapter also discusses supported logical architectures by JD Edwards, lists the customization options and limits, and offers guidance to those of you who are uncertain about your requirements.

## 3.1 Key logical architecture issues for Technology Foundation

The supported Technology Foundation logical architectures supported by JD Edwards differ primarily in two key dimensions: fault tolerance and security. The following sections discuss key characteristics of fault tolerance and security. They also outline some of the common mistakes that are made when defining requirements in these areas.

### 3.1.1 Fault tolerance

Any enterprise applications that a company relies on for day-to-day business can only be responsibly implemented with fault tolerance. Fault tolerance is achieved by applying physical and logical redundancy for all components in the topology. It is not simply turned on or off at a single layer of the architecture. It must span the architecture, and failover techniques vary in degrees of robustness and performance.

It is a common mistake to underestimate the cost to the business when an enterprise application is not available. This misperception occurs because the costs are often unseen, immeasurable, indirect, or simply forgotten. To measure these indirect costs, answer the question, “When the enterprise application is available for a given measurement of time...”:

- ▶ How much work will be performed when the application is operational?
- ▶ How many customers will place orders in this time frame?
- ▶ What is the price incurred to pay idle workers?
- ▶ How many customers will turn to a competitor as a result of the downtime?

Often, no one can say for sure. See the decision making guide in 3.4, “Selecting appropriate fault tolerance and security” on page 39, for techniques to more accurately answer these questions.

In the following sections, some of the key dimensions that define the quality of failover—degraded verses non-degraded failover, high verses continuous availability, and disaster recovery—are discussed. These dimensions are used to define the degree of failover for the logical architectures in this redbook. Most systems use varying degrees of failover at different levels of the architecture while adjusting to cost and risk.

#### Degraded verses non-degraded failover

Consider two machines that both receive work, which is dispatched from a workload manager as shown in Figure 3-1.

If Machine A fails, Machine B handles Machine A's work. However, it's important to realize that Machine B must now handle twice the workload it normally handles. Here, failover is said to operate in a *degraded mode* because performance is significantly affected.

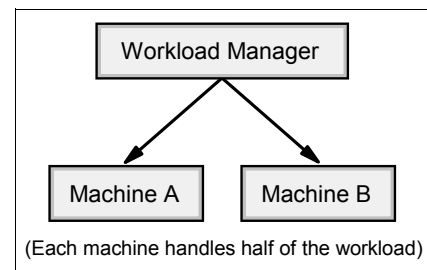


Figure 3-1 Distributed workload



By contrast, consider two other machines also configured so that Machine C can cover for Machine D as shown in Figure 3-1.

In this configuration, Machine D remains idle during normal operation. If Machine C fails, Machine D takes over but it does not have to handle twice the workload it normally does, as in the case of Machine A and B. Here, failover is said to operate in a *non-degraded mode*.

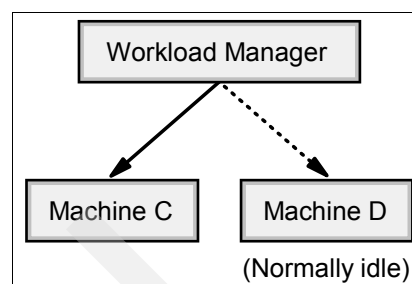


Figure 3-2 Nondistributed workload

**Note:** The IBM On Demand Business offering provides sophisticated mechanisms to ensure sufficient capacity exists regardless of failovers or unexpected performance spikes. You can learn more about this offering on the Web at:

[http://www.ibm.com/services/ondemand/start\\_overview.html](http://www.ibm.com/services/ondemand/start_overview.html)

A degraded failover configuration has the advantage of more efficient resource utilization and a better cost to performance ratio during normal operation. Non-degraded failover has the advantage of guaranteeing acceptable performance during a failover. By default, Technology Foundation is typically established with the sets of degraded and non-degraded failover identified in Table 3-1.

Table 3-1 Degraded and non-degraded sets

<b>Degraded failover</b>	HTTP Servers
	Portal Server
	Collaborative Server
	EnterpriseOne Web Server
	WebSphere Application Server
<b>Non-degraded failover</b>	IBM Directory Server
	Lotus Instant Messaging and Web Conferencing (Sametime)
	Lotus Team Workplace (QuickPlace)

For example, when an HTTP server fails in Technology Foundation, the remaining HTTP servers pick up the workload for the down machine. However, if a Lotus Team Workplace component fails, an idle Lotus Team Workplace component takes over, and the load on the newly operational machine does not exceed the load in normal operation.

In general, Technology Foundation HTTP servers have an great amount of excess capacity, so degraded failover does not present a problem. The remaining components that operate in degraded failover may or may not have sufficient excess capacity to compensate during downtime. The excess capacity of these components varies with the unique workload properties of the client and is determined during the sizing process.

**Note:** Any of the Technology Foundation components normally fielded with degraded failover can be upgraded to non-degraded failover simply by adding idle hardware to the infrastructure. These issues are determined during the sizing process that follows the logical architecture selection.

## High availability versus continuous availability

In its simplest form, fault tolerance or *high availability* consists of two machines with matching software. However, it is a mistake to assume that the system will always be available with this two-machine configuration. In Figure 3-3, Machine A's failure leaves the system in a vulnerable state because Machine B must handle the workload for the enterprise with no backup.

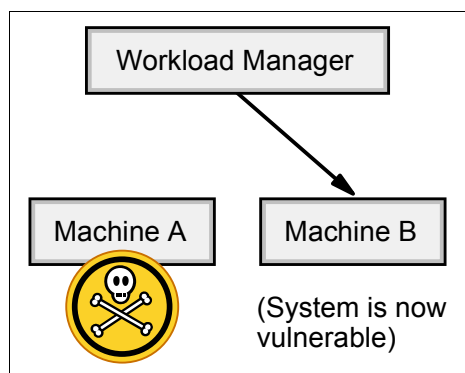


Figure 3-3 High availability mode works

If Machine B becomes unavailable, the entire system becomes unavailable (Figure 3-4). Similarly, this state occurs during any maintenance operation such as an upgrade to the operating system or applying an operating system fix that causes Machine A to become unavailable.

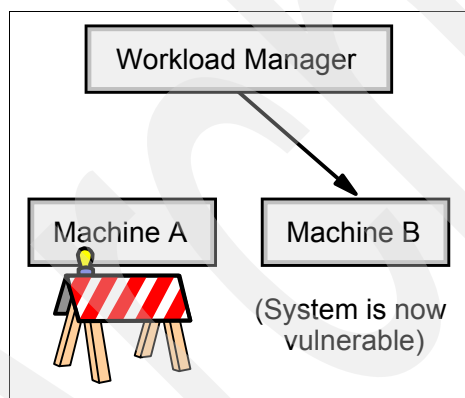


Figure 3-4 One machine becomes unavailable

If a failure occurs during a planned outage, the Workload Manager is unable to dispatch requests. Also, the enterprise application becomes unavailable to all users even though the system is defined to be highly available. See Figure 3-5.

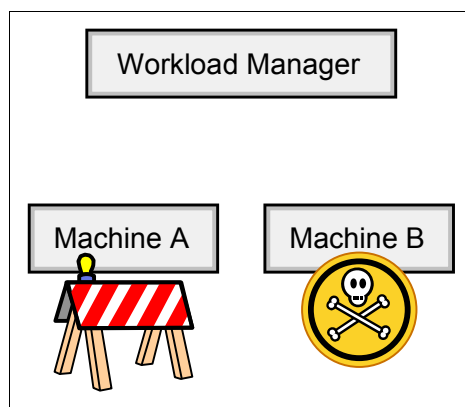


Figure 3-5 System becomes unavailable

This vulnerability can be solved with the addition of another machine as shown in Figure 3-6. With the addition of another machine, the system is now said to be *continuously available* because it can withstand a failure during an upgrade operation.

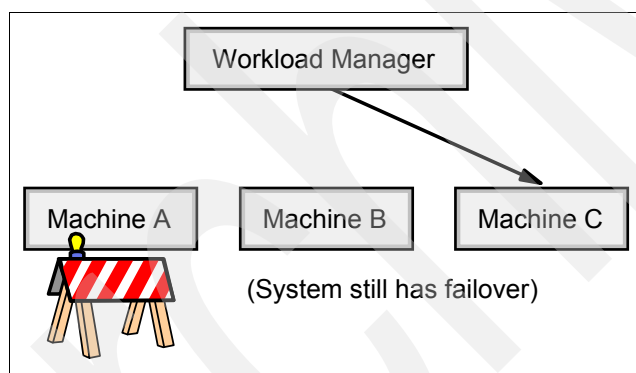


Figure 3-6 One of three machines becomes unavailable

**Note:** iSeries servers can provide service even through operating system upgrades.

See “The business case for high availability” on page 39 to understand these concepts.

Continuous availability can only be achieved with a minimum of three machines. Systems that implement failover with more than three machines are also referred to as *continuously available*. As additional machines are added, the probability of an unplanned outage progressively declines.

**Note:** As servers are added, the probability of a failure on any one server in the system increases since there are simply more servers. However, when the servers operate independently from each other, the probability of a system-wide failure declines. This combined probability can be calculated given the mean time between failures. This varies significantly across operating systems and hardware platforms. For example, iSeries servers provide a longer mean time between failures than PCs.

Under certain circumstances, even with continuous availability, the system is still vulnerable. If a failure occurs during a planned upgrade, the system is again vulnerable to a system-wide failure since a single machine is processing all requests with no backup (Figure 3-7).

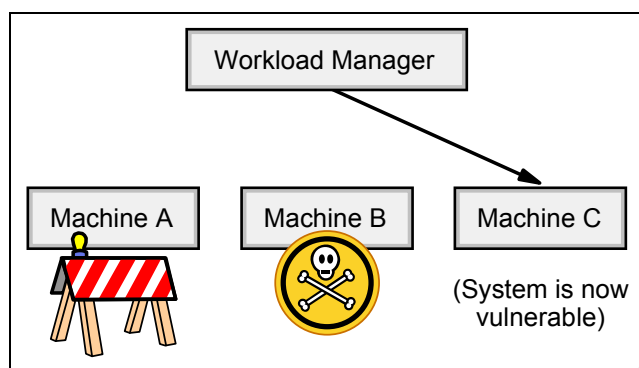


Figure 3-7 Continuously available workload during planned upgrade

As the system is horizontally scaled, the probability decreases for the enterprise applications to become unavailable for all users.

**Note:** To determine the appropriate failover strategy for a given business, compare the cost of adding and maintaining additional machines against the estimated cost of the enterprise applications to be unavailable to all users. The solution varies by the demands and requirements of each business. See the decision making guide in 3.4, “Selecting appropriate fault tolerance and security” on page 39.

It is tempting to pronounce systems as highly available or continuously available simply by counting the number of machines and associating *highly available systems* with the pattern of doubling, and *continuously available machines* with the pattern of tripling.

But what if it takes more than a single machine to support the normal system load? In this case, a system with two machines cannot correctly be called highly available since the system denies transactions if either machine fails. Such an architecture cannot afford to lose a single machine. In this case, high availability can be achieved only with a minimum of three machines, which is normally associated with continuous availability. In turn, in this scenario, continuous availability requires a minimum of four machines.

**Note:** The IBM sizing methodology follows the logical architecture selection and resolves this issue. A discussion of this process is outside the scope of this redbook. Nevertheless, to preserve clarity in diagrams and concepts within this redbook, we represent high availability with a double configuration and continuous availability with a triple configuration.

## Disaster recovery

Unlike general fault tolerance where failover occurs within an installation or data center, disaster recovery involves a failover that operates in the face of a catastrophe such as the physical destruction of the data center building. Failover of this magnitude is often caused by fires, city power outages, network outages, or disasters of nature.

Disaster recovery is a form of high availability that provides redundancy for the data center itself. It is usually geographically distinct and far enough away from the production data center so that the same natural disaster does not affect both sites.

Different kinds of disaster recovery techniques are available. The quality of the appropriate method depends on the needs of the business. IBM Business Continuity Services (BCS) allows clients to select from disaster recovery offerings depending on how fast the business needs to be back in operation and running during a site failure. For example, one client solution may be to dynamically mirror a data center. This solution can provide continuous availability and completely uninterrupted service during failure at the business site. A different customer may maintain a failover site that simply keeps relatively current with information stored at the primary center, thereby providing backup service after recovering from a brief service interruption.

**Important:** A common yet disastrous mistake is to assume that a data center has a failover mechanism simply because the electronic information is backed up to magnetic media and stored off site.

Backing up a system to removable media and storing the media off site does not constitute disaster recovery. When catastrophe strikes a data center, clients that incorrectly assumed that they are protected are now struck by the realization that their preserved data is not usable without an environment on which to run it.

Systems requiring true disaster recovery tend to be large deployments. They usually require customization and special treatment for far more systems than just Technology Foundation and EnterpriseOne. Disaster recovery is more appropriately addressed when discussing failover strategies for the EnterpriseOne server itself. While certainly worth mentioning, the topic of disaster recovery is outside of the scope of this redbook.

**Note:** IBM business continuity and recovery services are devoted to handling the challenging problems of protecting critical business infrastructure. See the following Web site for more information about these services:

<http://www.ibm.com/services/continuity/recover1.nsf/documents/home>

### 3.1.2 Degree of security

As with fault tolerance, security is not a property that is simply *on* or *off*. There are degrees of security.

While most companies distrust the Internet, most corporations treat their internal network as completely trusted. They feel that as long as they are protected from the Internet, they are safe. This is a potentially disastrous assumption since the vast majority of security compromises occur from within the corporate network itself.

Any enterprise application (and its Web front-end components) that is not protected from the internal corporate network can expect problems. This is especially true if any of its components reside on commonly attacked PC operating systems.

Technology Foundation offers two levels of security that both provide a responsible yet increasing level of protection. The levels are *standard* and *highly secure*.

## Standard security logical architecture

A standard security Technology Foundation logical architecture is characterized by a demilitarized zone (DMZ) that protects enterprise applications from the untrusted internal, corporate network. HTTP servers reside in the DMZ and serve non-sensitive static data. The HTTP servers pass requests for dynamic information through a firewall that guards the internal components. Figure 3-8 shows a *Standard Security, Highly Available* configuration.

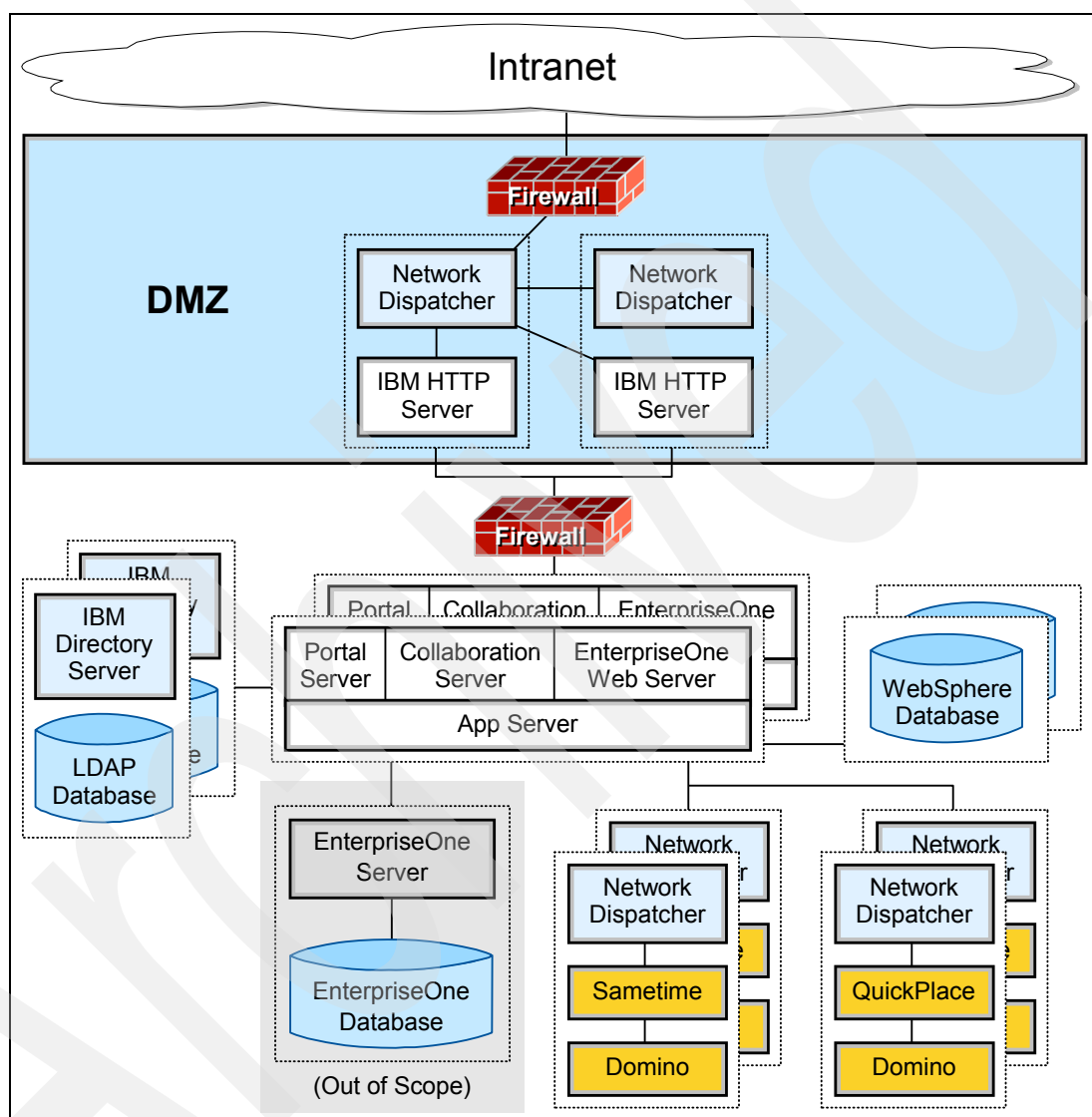


Figure 3-8 Logical architecture with standard security: Load balancers and HTTP in DMZ

The two standard security logical architectures provide the minimal amount of security that can reasonably be considered responsible and appropriate for enterprise applications. Outside untrusted machines can talk to machines in the DMZ, and internal enterprise application machines trust messages that originate from within the DMZ. Because of this, machines in the DMZ act as intermediaries. The DMZ allows untrusted outside machines to indirectly talk to internal trusted machines.

**Note:** Translating the logical DMZ into an actual physical realization is outside of the scope of this redbook. Refer to the sizing process described in 2.1, “Methodology for producing a physical architecture” on page 8.

The introduction of a DMZ into the logical architecture causes some challenges later when the logical architecture is transformed into the end physical architecture. For example, an appropriately sized server may be able to handle the performance requirements for all the Technology Foundation components. It is tempting to place the DMZ components into one logical partition (LPAR) of the system and the remaining components in another partition, with software or hardware firewalls separating them. However, components inside of the DMZ cannot be hosted on the same physical system as components in the internal trusted area.

**Important:** The idea of having a DMZ is to physically isolate machines into a semi-trusted zone. Never use a single machine to host both the DMZ and the internal components.

Some Java 2 Platform, Enterprise Edition (J2EE) architects prefer to physically divide Web content servers from Enterprise JavaBean (EJB™) servers, with servlets and JavaServer™ Pages™ (JSPs) in the DMZ while protecting EJBs behind the firewall. While this approach is defensible, the application server is required to traverse physical machines several times over the course of a single user's transaction and degraded performance results. Since security, installation, and supporting administration scripts are complicated with little return in benefits, this variation is not a supported customization.

### Highly secure logical architecture

The highly secure logical architecture (Figure 3-9) is distinguished from the standard security architecture by the introduction of reverse proxy elements.

IBM Tivoli® Access Manager component, *WebSEAL*, is a multi-threaded reverse proxy. WebSEAL serves as Technology Foundation's mechanism to force users to authenticate at the most forward located components of the architecture. Users who are unable to provide appropriate login and password credentials cannot even begin to communicate with machines in the enterprise, even indirectly through DMZ intermediaries. WebSEAL front ends back ends Web services by applying a security policy to protected internal objects. It can provide single signon (SSO) solutions and incorporate back-end Web application server resources into its security policy. Founded on the IBM HTTP server, it functions by listening to the typical HTTP and Secured HTTP (HTTPS) ports.

Only packets identified as being from the reverse proxy machines are allowed into the internal networks. DMZ proxies always act on behalf of the outside users.

To keep their purpose clear, Figure 3-9 shows two internal firewalls. There is a third in the DMZ. However, many firewalls provide the ability to link more than two networks together. It is certainly possible that these two logically distinct firewalls can be combined into a single physical firewall for the actual deployment.

While hardware selection is performed in the sizing process, some hardware platforms are inherently more secure than others. For example, the iSeries OS/400 operating system uses a powerful and unique object paradigm to secure system resources. Since files, TCP/IP sockets, Java classes, and virtually all resources appear as objects, the system is better secured against buffer overrun attacks, for example. In addition, database tables, user sessions, executables, files, and other sources of security concerns are kept safely in a single-level storage virtual address space. Refer to *IBM @server i5 and iSeries System Handbook*, GA19-5486, and for a description of the unique iSeries secure architecture, see:

<http://www.ibm.com/servers/enable/site/porting/iseries/overview/overview.html>

Resource users are prevented from writing beyond the boundaries of objects. Since higher end operating systems are more expensive and less common, there are fewer and less knowledgeable attackers.



**Note:** In practice, hardware selection is frequently known from the onset and is often determined by corporate standards and the client's corporate technology roadmap. The sizing process often merely validates that the desired platform is capable of supporting the anticipated system load.

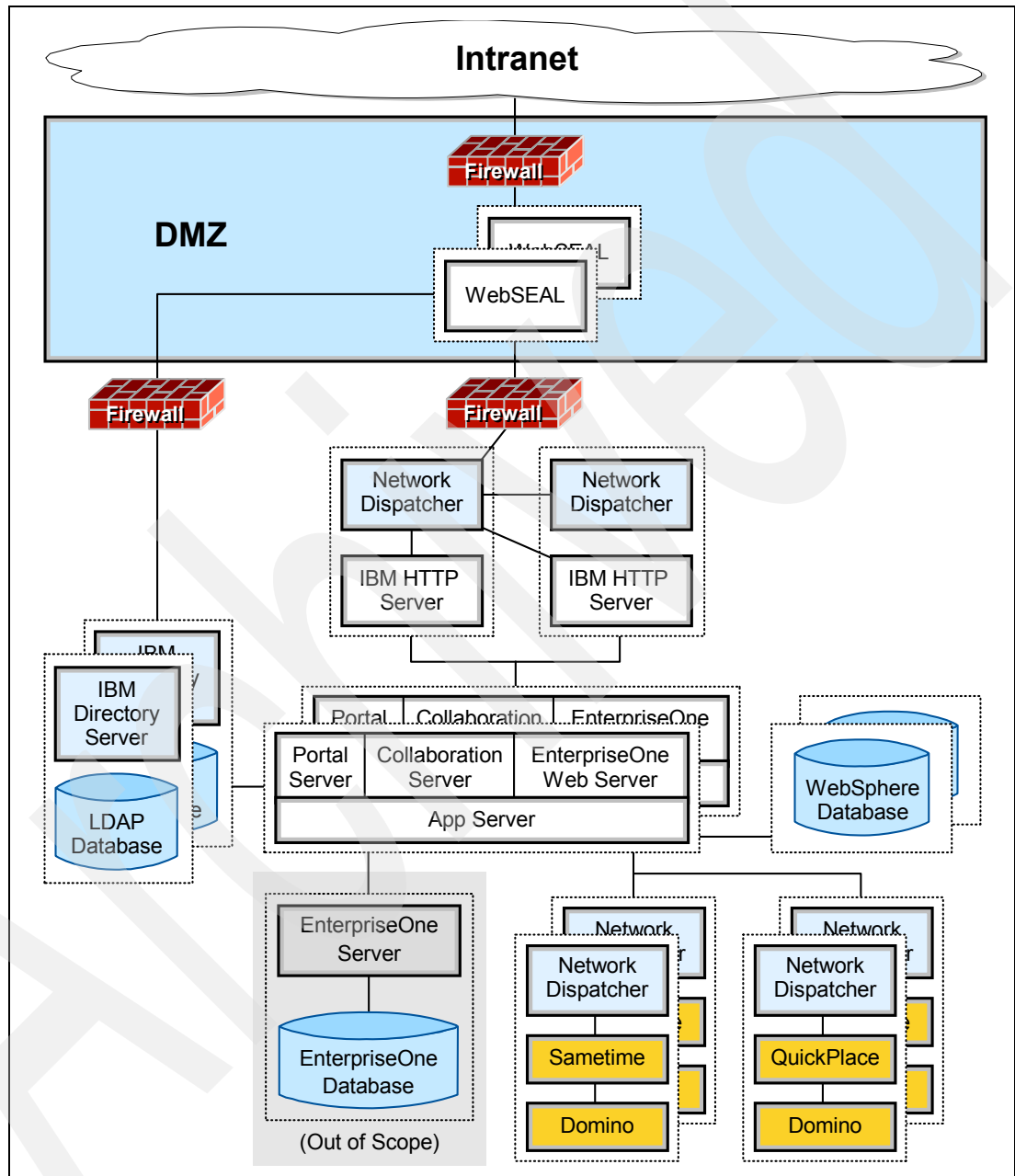


Figure 3-9 Highly secure logical architecture: Reverse proxy in DMZ



## 3.2 Supported, standard logical architectures

This section provides a comprehensive list of the four supported logical architectures for Technology Foundation. Each architecture can be altered. However, to remain supported, the alterations must remain within the guidelines cited in 3.3, “Allowed customizations to logical architectures” on page 29. You must select a single logical architecture from this supported list. Table 3-2 summarizes the supported logical architecture options.

Table 3-2 Supported logical architecture options

Availability	Standard security	Highly secured
Highly available	The minimal level of fault tolerance that can responsibly service an enterprise application	The minimal level of fault tolerance that can responsibly service an enterprise application
	The minimal level of security that can responsibly protect an enterprise application	Solid protection from external and internal corruption
Continuously available	Fault tolerance sufficient to provide service if a single failure occurs during a planned upgrade	Fault tolerance sufficient to provide service if a single failure occurs during a planned upgrade
	The minimal level of security that can responsibly protect an enterprise application	Solid protection from external and internal corruption

Each of these standard architectures is ultimately mapped into a physical architecture. For example, there is a physical architecture for the *Standard Security, Highly Available* logical architecture for the xSeries, pSeries, and iSeries servers. As new versions of EnterpriseOne and Technology Foundation are produced, they are tested on these platforms.

Oracle uses IBM hardware and software to test base configurations. Non-standard hardware or software can significantly reduce performance or result in unique problems. For example, the tight hardware binding between the xSeries and the iSeries servers is not realized with standard Intel hardware.

The four possible architecture options are described in the following sections. These four options apply both to Technology Foundation Versions 4 and 5. With Version 5, an additional machine is required to host WebSphere Network Deployment Manager. To save space and to ensure the diagrams apply to Version 4, the Network Deployment Manager is not shown.

**Important:** Technically WebSphere Network Deployment Manager introduces a single point of failure into the architecture since it is not a redundant component. However, this is rarely an issue for most Technology Foundation clients. That is, node agents immediately use the latest change from the Deployment Manager after it is recovered and client environments tend to evolve slowly after the initial deployment. For further information, see the “Deployment Manager Failures” chapter in *IBM WebSphere V5.1 Performance, Scalability, and High Availability: WebSphere Handbook Series*, SG24-6198.

### 3.2.1 Standard Security, Highly Available

Figure 3-10 shows the *Standard Security, Highly Available* logical architecture. This logical architecture offers:

- ▶ The minimal level of security that can responsibly protect an enterprise application
- ▶ The minimal level of fault tolerance that can responsibly service an enterprise application

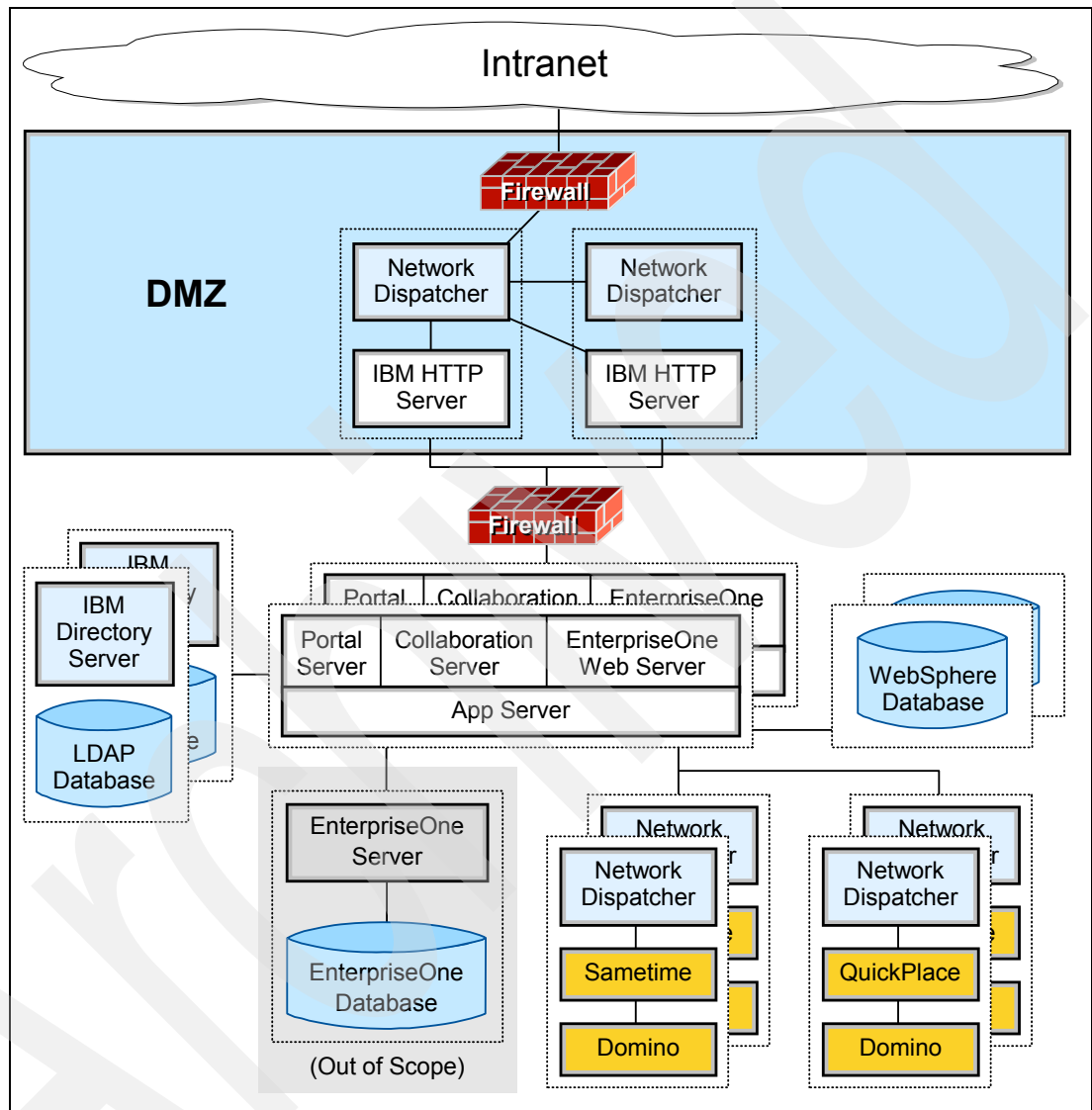


Figure 3-10 Standard Security, High Availability logical architecture

### 3.2.2 Standard Security, Continuously Available

Figure 3-11 shows the *Standard Security, Continuously Available* logical architecture. This logical architecture offers:

- The minimal level of security that can responsibly protect an enterprise application
- Fault tolerance sufficient to provide service if a single failure occurs during a planned upgrade

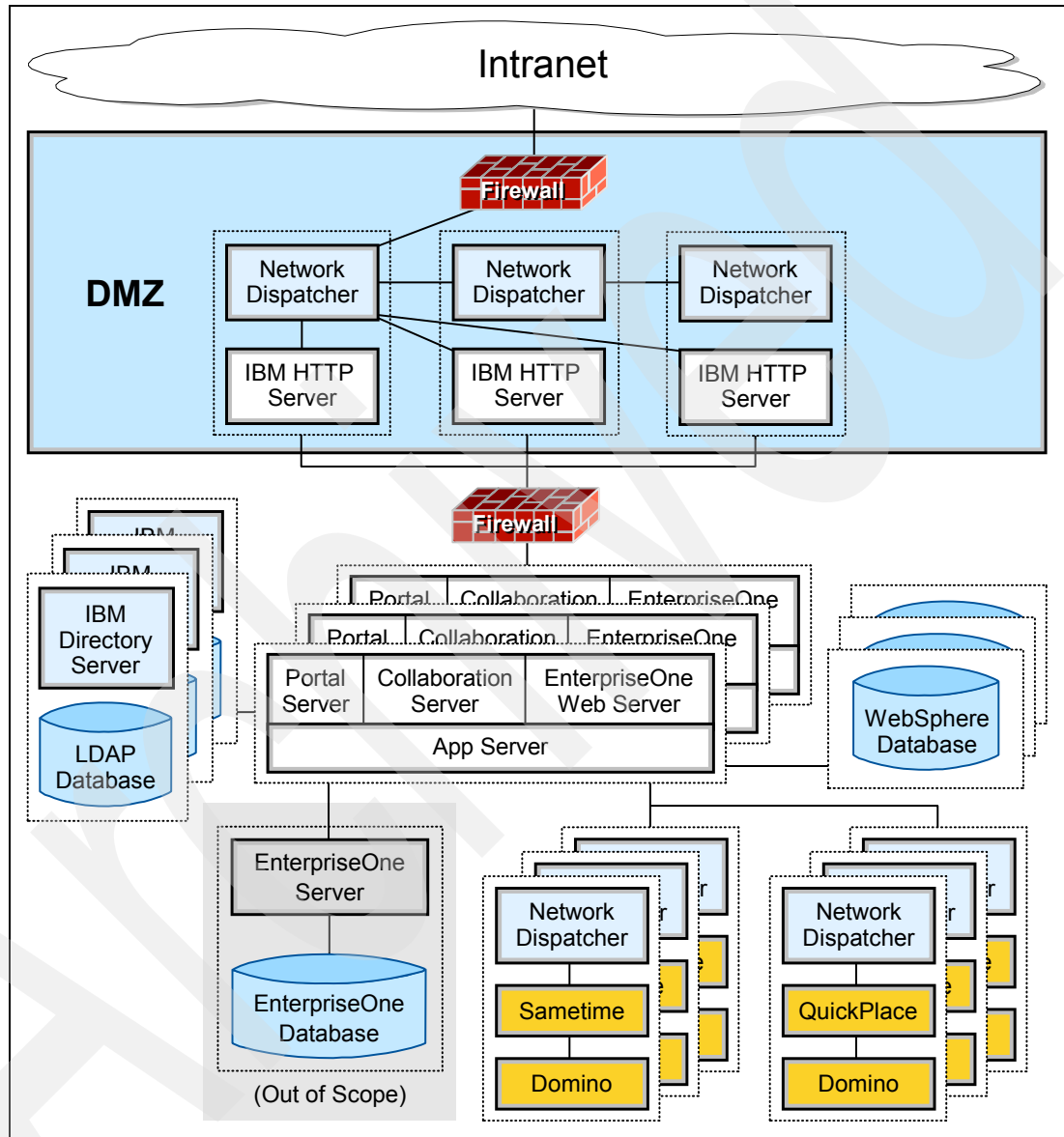


Figure 3-11 Standard Security, Continuously Available logical architecture

### 3.2.3 Highly Secure, Highly Available

Figure 3-12 shows the *Highly Secure, Highly Available* logical architecture. This logical architecture offers:

- Solid protection from security compromises
- The minimal level of fault tolerance that can responsibly service an enterprise application

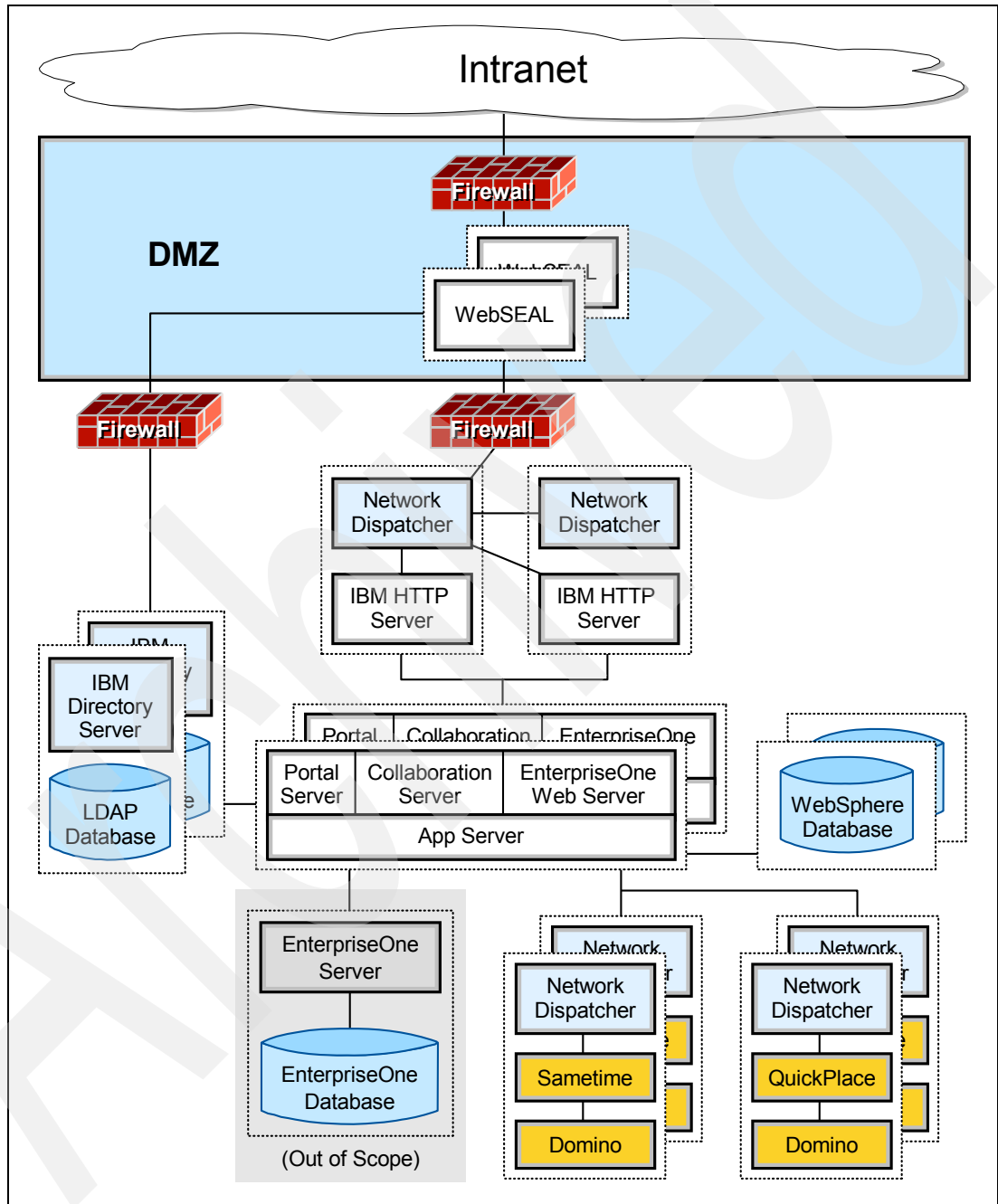


Figure 3-12 Highly Secure, Highly Available logical architecture

## Highly Secure, Continuously Available

Figure 3-13 shows the *Highly Secure, Continuously Available* logical architecture. This logical architecture offers:

- Solid protection from security compromises
- Fault tolerance sufficient to provide service if a single failure occurs during a planned upgrade

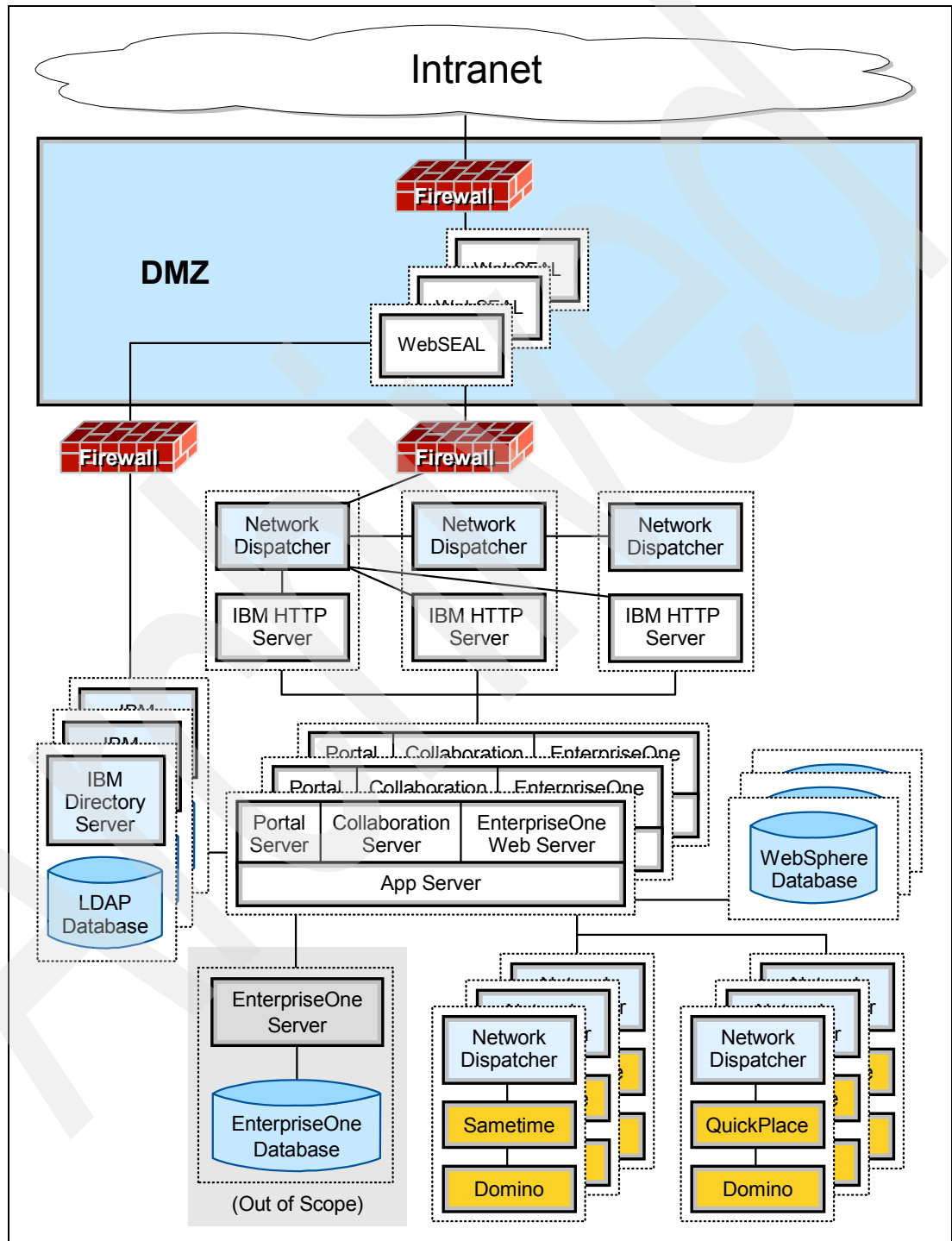


Figure 3-13 Highly Secure, Continuously Available logical architecture

### 3.2.4 Test architecture

Enterprise architectures necessarily require high availability because the cost of enterprise downtime vastly exceeds the cost of downtime for normal corporate machines. Similarly, enterprise applications must be far more secure than the average corporate machines since they tend to manage sensitive data. Nevertheless, there is a clear need for a minimal deployment architecture to support proof of concept, testing, and pilot projects. Figure 3-14 illustrates the architecture as intended to address the needs of such environments.

This architecture provides little security and no availability. It places far more load on a single machine than is ever prudent to do in a production environment. However, it is sufficient to establish an environment capable of processing a light transaction load and to demonstrate display capabilities. It can also help to debug non-performance related problems.

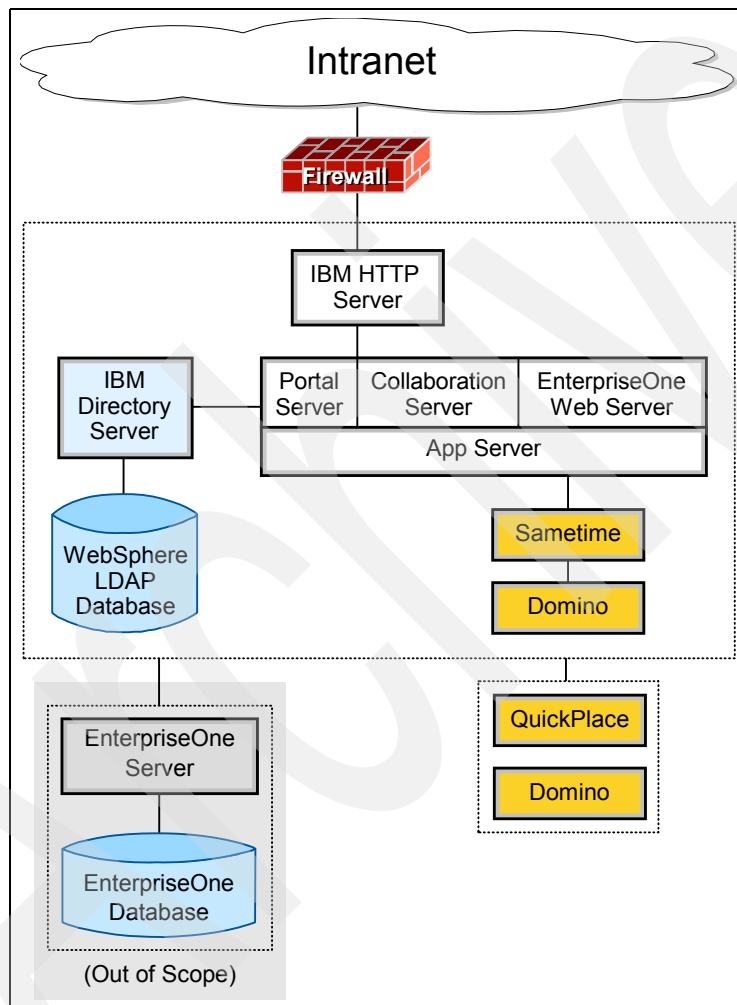


Figure 3-14 Test architecture for pilot projects

## 3.3 Allowed customizations to logical architectures

There are limitless ways to successfully organize and deploy the hardware and software for Technology Foundation. For example, HTTP servers may reside on the same machines as the WebSphere server. The databases can be consolidated or run separately. In a pSeries environment, the Web application servers can reside on separate LPARs or on a single LPAR.

**Application server:** In this redbook, the term *application server* is meant to refer to the WebSphere Web application server and is accepted J2EE parlance. This contrasts with some JD Edwards documents that use the term *application server* (or *app server*) to refer to the back-end EnterpriseOne server's kernel processes.

Such architectures are often functionally equivalent. However, they have different performance characteristics, require different support scripts, have different interfaces to each other, and exhibit different problems. Allowing every client to have its own custom architecture creates an intractable support problem for Oracle while adding no additional value. An overabundance of supported architectures invisibly detracts from the quality and reliability of the software for all end users by inefficiently consuming resources.

Consequently, Oracle and IBM limit the selection of supported logical architectures, bounding a finite set of options, which can be thoroughly tested and easily supported. Some customizations are allowed and may be necessary for these architectures. For example, most organizations already have a Lightweight Directory Access Protocol (LDAP) directory service in place. They do not want to establish an additional new one even though the default architectures require one inside the Technology Foundation environment. Such exceptions are practical and necessary.

The following sections define the limited set of acceptable customizations.

**Note:** Horizontal and vertical scaling are not considered to be variants of logical architectures. The scaling process is a variant of the sizing process that follows the logical architecture selection and consequently is out of the scope of this book.

### 3.3.1 Connecting to third-party external directory servers (LDAPs)

By default, each logical architecture comes with an IBM Directory Server as the LDAP platform. For example, the Technology Foundation Domino Server cluster can also be used as a directory sever. However, if an EnterpriseOne client already has a corporate LDAP structure, in most cases, it is desirable to use the existing LDAP rather than to establish a new one. Microsoft's Active Directory, Sun's Java System Directory Server, Lotus Domino Enterprise Server, and other LDAP servers can be used as information directories.

Using an alternate LDAP platform is a supported variation of any of the logical architectures. For example, the Standard Security, Highly Available logical architecture can be modified as shown Figure 3-15.

This architecture differs from the Standard Security, Highly Available logical architecture simply because an external directory server is used for validation. Here, the LDAP server is shown to reside inside the secured EnterpriseOne network. However, it may reside outside of the DMZ in the untrusted corporate network where encrypted communication is especially warranted.

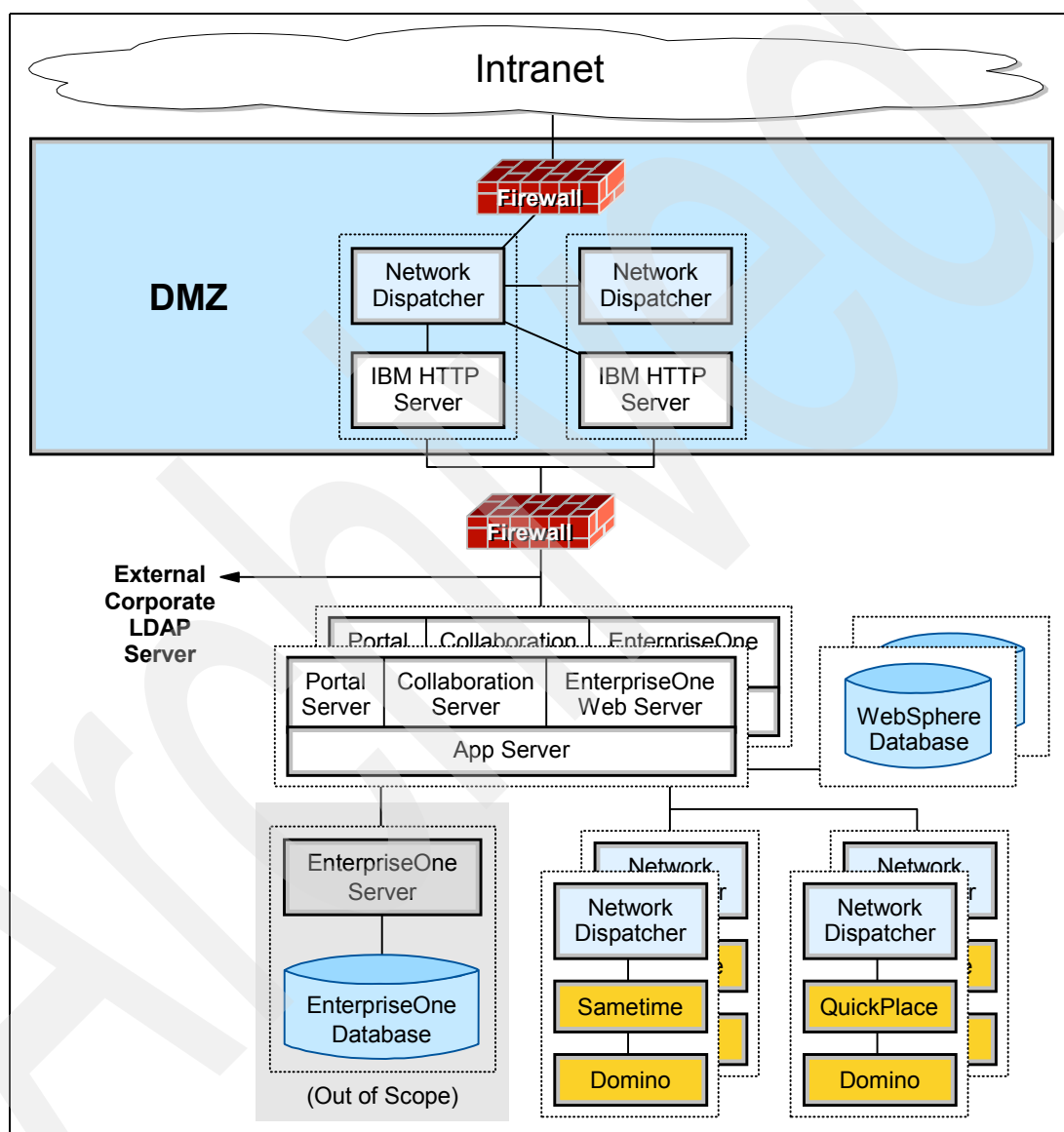


Figure 3-15 Logical architecture variation: External corporate directory server



### 3.3.2 Downgrading availability for non-mission critical elements

High availability is necessary only for the elements of a system where the cost of downtime exceeds the cost of establishing and maintaining high availability. For a true enterprise application, this is normally the entire system. However, some clients may not rely on Lotus Instant Messaging and Web Conferencing and Lotus Team Workplace for their mission-critical operational needs. Consequently those components may be configured without high availability as shown in Figure 3-16.

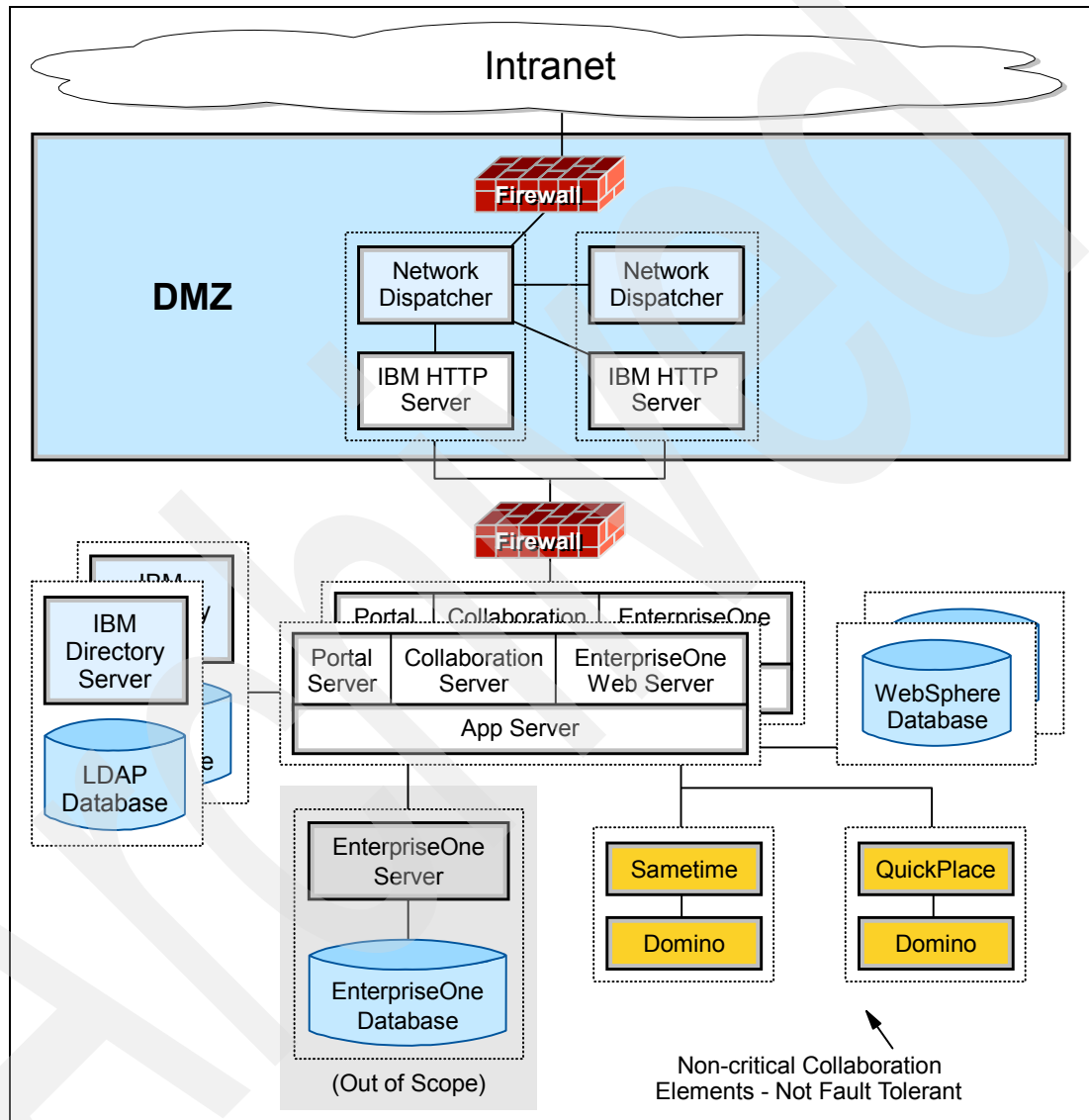


Figure 3-16 Logical architecture variation: No redundancy in non-critical elements

Similarly, some clients that require continuous availability may only require high availability for the Lotus collaboration portion of Technology Foundation. This configuration is shown in Figure 3-17.

Downgrading the availability of Lotus collaboration components is an acceptable variation of the Technology Foundation logical architectures.

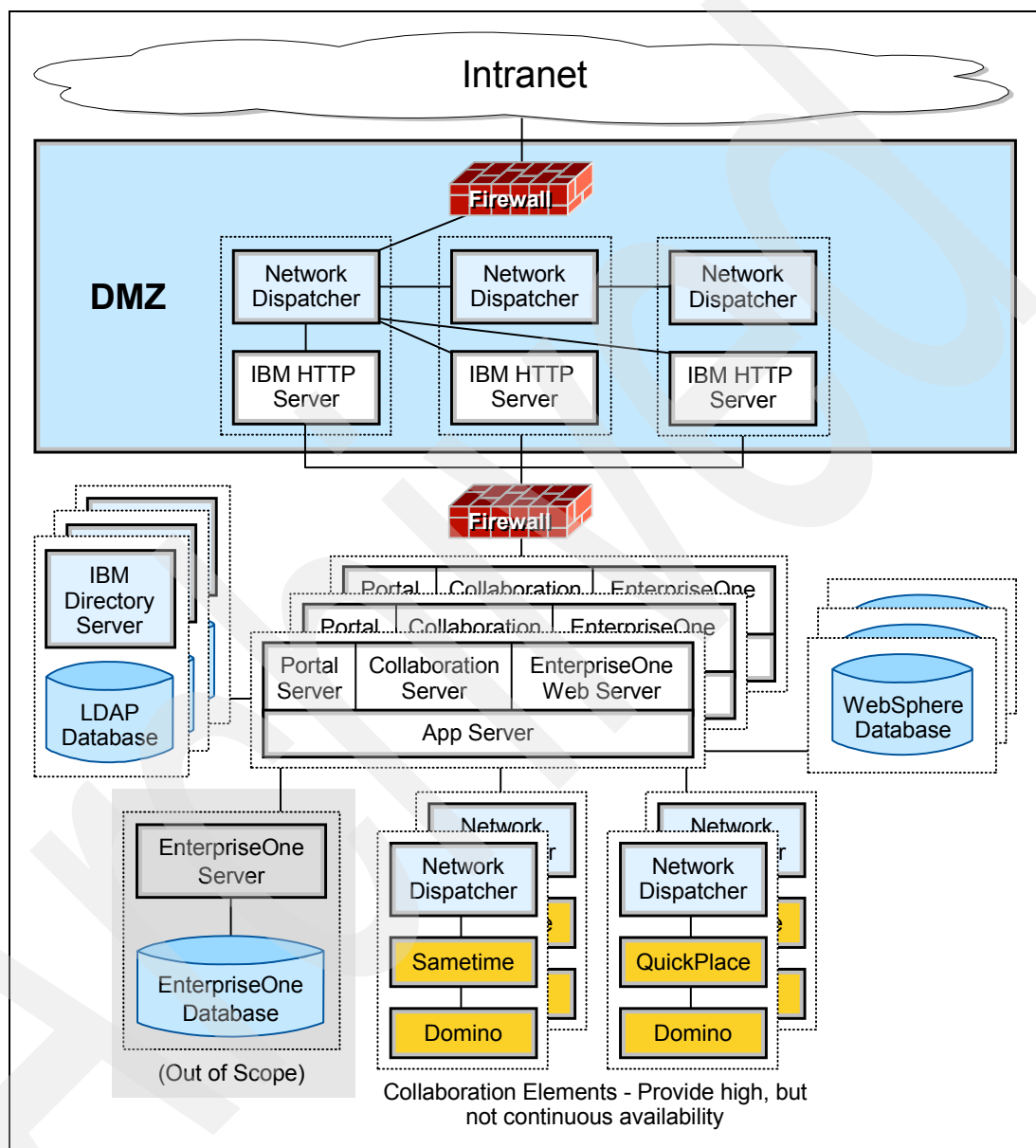


Figure 3-17 Logical architecture variation: Downgraded availability for non-critical components

### 3.3.3 Increasing performance for remote locations

Corporations often provide access to enterprise applications in remote locations over slower network connections such as T1 or integrated services digital network (ISDN) lines. Consequently, these remote locations can suffer in performance as a result of the latency incurred when accessing remote information. IBM provides WebSphere Edge Server for these situations.

**Note:** More precisely, WebSphere Edge Server is composed of two subproducts: *Web Traffic Express* and *Network Dispatcher*. The Web Traffic Express portion of Edge Server aids performance by caching static content for remote locations.

Edge servers cache static content and locate the information geographically closer to the remote location. Slow access latencies are avoided and performance can be substantially increased with such caching. Figure 3-18 shows the Standard Security, Continuously Available logical architecture augmented with Edge Servers.

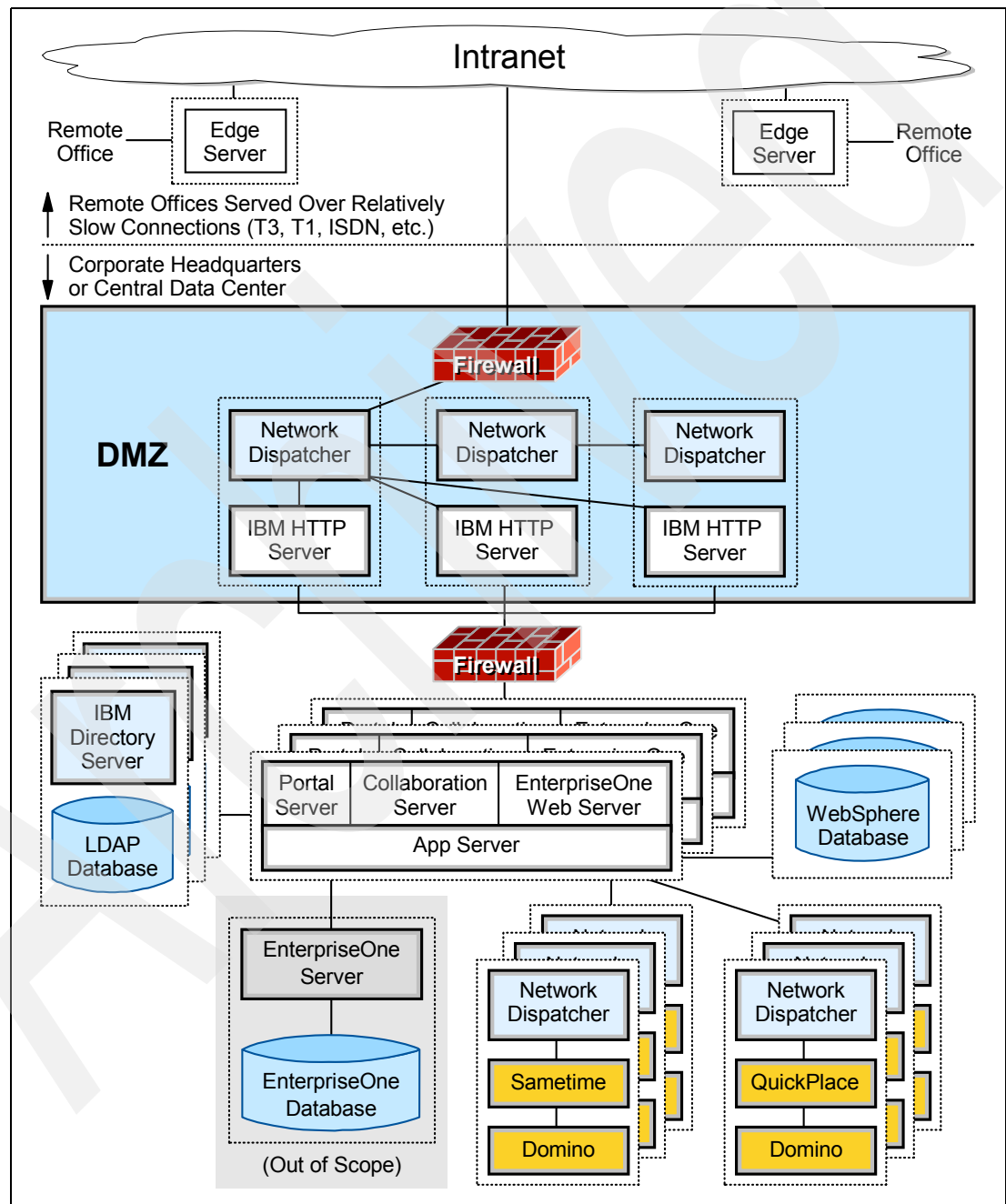


Figure 3-18 Logical architecture variation: Increasing performance with Edge Servers

Edge Server documentation most commonly shows caching components deployed between the Network Dispatcher and the IBM HTTP Server (not shown here). This is because, for Technology Foundation clients, the end users normally reside directly on the high performance internal network. The enterprise application typically does not have an abundance of static content to push out to tens of thousands of users at numerous disparate locations across the Internet.

Inserting Edge Server caching components and forward locating cached information closer to users who are already on a high-speed internal network has little effect on performance and the added complexity is not warranted here. Consequently the Edge Server is absent from its normally depicted location in this logical architecture. However, performance for Technology Foundation end users who are located at remote facilities can be significantly increased by the presence of Edge Server at the remote location. This is especially true if the link between the corporate network and the remote location is relatively slow, such as through an ISDN line or T1 connection.

For a description about how you can use Edge Server's Web Traffic Express caching component to optimize performance at remote branch offices, see Chapter 2 in the IBM Redbook *WebSphere Edge Server: Working with Web Traffic Express and Network Dispatcher*, SG24-6172.

### 3.3.4 Very high internal security

For environments that are prone to internal virus infection, it may be desirable to compartmentalize the Technology Foundation components for extra security by adding additional firewalls between layers of the architecture. Figure 3-19 shows how to upgrade the security for the Highly Secure, Highly Available logical architecture.

This very Highly Secure, Highly Available logical architecture offers solid protection from internal (and external) corruption since components are protected from corruption on other components. For example, application servers are protected from the Lotus Collaboration elements by a firewall.

Several of the firewalls shown in Figure 3-19 can be combined into a single piece of hardware. Many firewalls provide the capability of mapping between more than two Internet Protocol (IP) address spaces.

For example, the two firewalls that stand in front of the EnterpriseOne server and the Lotus Instant Messaging and Web Conferencing and Lotus Team Workplace components can potentially be combined into a single firewall depending on the type of hardware selected. Similarly, the firewalls that connect WebSEAL to the Network dispatcher and WebSEAL to the IBM Directory Server can potentially be combined. The remaining firewalls have opportunities for physical architecture consolidation as well. They are separated in the logical diagram to show which component is being protected from which other component and to clarify how they must eventually be configured.

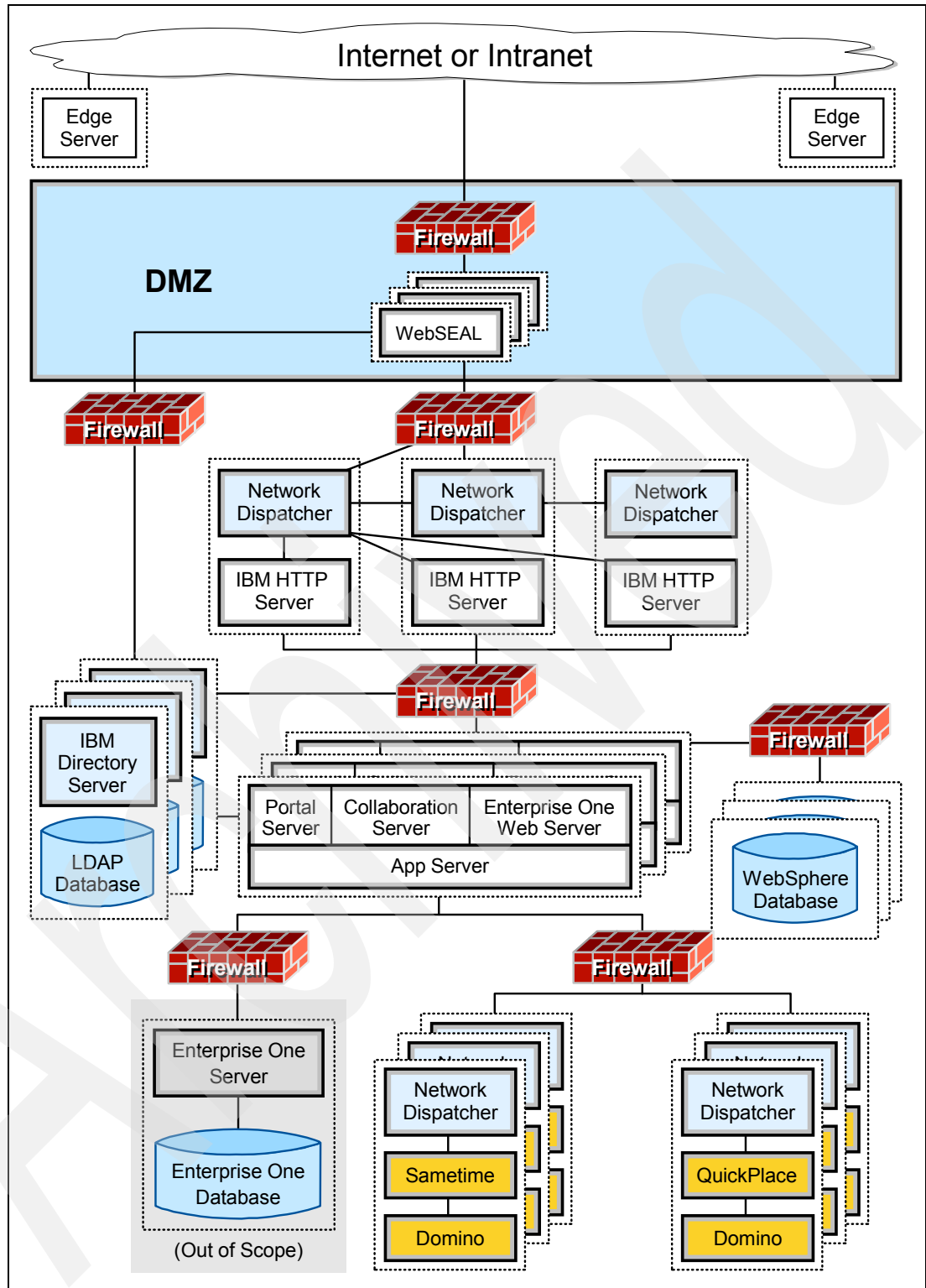


Figure 3-19 Logical architecture variation: Very highly secure

### 3.3.5 Providing Internet access

Public access across the Internet allows enterprise applications to be accessible for employees to log in and work from home or to provide customers with access. However, public access warrants one of the highest levels of security, particularly if sensitive information is to be made accessible. Companies that provide Internet access often come with continuous availability requirements. The architecture shown Figure 3-20 augments the very Highly Secure, Continuously Available environment with Edge Servers to enhance performance.

The Edge Server components in this architecture are geographically distributed at strategic locations across the Internet. They are intended to cache static content and place that content in closer proximity to groups of users that suffer from poor performance. Some analysis of the geographic patterns of use must occur to determine where to most effectively place these machines.

This performance optimization is commonplace on the Internet. By selecting *view source* from a browser that is displaying virtually any popular Web site, careful analysis of the Uniform Resource Locators (URLs) in the Hypertext Markup Language (HTML) reveals that the page is actually assembled from numerous sites. To increase performance, these sites are often selected based on the geographic location of end-user's browsers.

This architecture is secured by layers of firewalls and reverse proxy mechanisms. It further decreases security risks by protecting the internal machines from each other. For example, in the standard or highly secure architectures, if a Web Application Server machine obtains a virus, there is nothing to prevent that virus from spreading to the Lotus Instant Messaging and Web Conferencing and Lotus Team Workplace machines.

A common mistake is to assume that internal machines simply aren't susceptible to viruses since they are protected by so many layers at the front of the architecture. While it's true that viruses are far less likely to penetrate these layers, the added protection is often irrelevant. Viruses typically infiltrate enterprise systems by being invited in to the corporate network. For example, a system administrator may receive an infected e-mail on a machine that resides on the protected network and has access to critical enterprise machines. Similarly, an administrator may bring in an infected mobile computer from home and connect it to the internal network.

For the same reasons discussed in the previous section, several of the firewalls in this diagram can be optionally combined into a single piece of hardware.

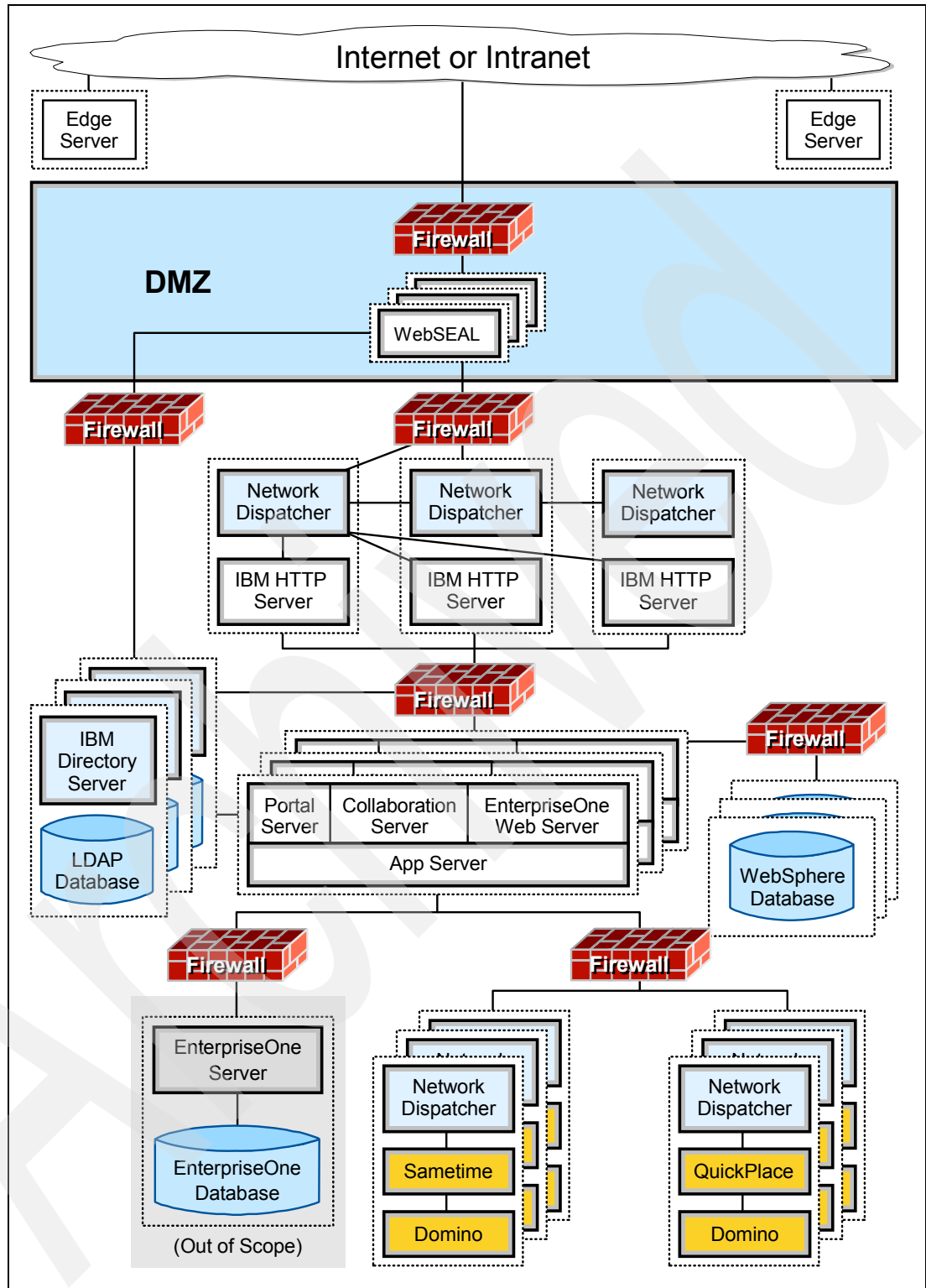


Figure 3-20 Logical architecture variation: Very highly secure with Internet-ready performance

### 3.3.6 Secure Sockets Layer accelerators

Encrypting information is generally a good idea in either the standard or highly secure Technology Foundation logical architectures. However, encryption, when performed by software, can be accompanied by a performance hit.

*Secure Sockets Layer (SSL) accelerators* are hardware devices inserted just before the Web server that performs the computation intensive step of decrypting encoded information. SSL has occasionally been identified as a bottleneck for some customers, particularly if they are running on PC platforms. SSL accelerators relieve these types of bottlenecks effectively and are a supported variation of all logical architectures.

SSL accelerators also serve to solve the problem of load balancing when using SSL. Ordinarily, load balancers do not support server affinity when SSL is enabled because the information necessary to support affinity is encrypted. By inserting an SSL accelerator before the load balancer, the information is decrypted prior to balancing. Also the dispatcher can ensure that requests from the same user are redirected to the same application server that previously serviced this user. Application server affinity dramatically increases the frequency of cache hits.

### 3.3.7 Third-party IP sprayers

Network Dispatcher is the software component that allows multiple HTTP servers to look like a single server to the outside world. As the user community grows, additional HTTP servers can be added to scale the system without disturbing the user community in any way.

Network Dispatcher performs the functions of load balancing and IP spraying in the supported logical architectures. However, alternate hardware or software sprayers can be used in place of Network Dispatcher. Simple Domain Name System (DNS) round robin sprayers can be used. However, such load balancers have distinct disadvantages, including:

- ▶ DNS sprayers frequently have partial or completely absent support for session affinity.
- ▶ Many DNS sprayers do not automatically recognize down machines in a cluster. Consequently, they may incorrectly route live requests to non-responsive servers.
- ▶ Since DNS sprayers are part of the DNS network, peer DNS servers may cache antiquated IP addresses. In the event of a machine failure, even if the local environment properly reroutes new requests to other servers, external DNS servers may not reflect the IP changes for some time.
- ▶ DNS sprayers tend to have weak monitoring support.

Third-party IP sprayers only make sense for clients with severe cost restrictions. This concern only applies to Technology Foundation Version 4. Beginning with Version 5, Network Dispatcher comes bundled with the other components.

### 3.3.8 Third-party database vendors

Clients may substitute a third-party database product for DB2 Universal Database (UDB), providing that database is in WebSphere's list of supported database vendors. For example, Microsoft and Oracle databases are possible substitutions. Moreover, the database is not required to run on a separate machine. Many clients, for example, have a large, well supported, mirrored, and high performance database running on the back-end iSeries server. This back-end database can be used instead of the various WebSphere and portal local database instances.



## 3.4 Selecting appropriate fault tolerance and security

The following sections provide guidance on how to select one of the supported logical architectures described in 3.2, “Supported, standard logical architectures” on page 23.

### 3.4.1 Determining fault tolerance requirements

Many enterprise solutions in the industry today exhibit single points of failure, perhaps at a PC-based firewall or IP sprayer. This is surprising because the business value received by customers from having redundancy is dramatically higher than the overhead cost of creating and maintaining redundancy.

Why would you even have such technologies as J2EE runtimes with all of the built-in qualities of service if a single weak link in the chain can still bring down the system? Why would you have autonomic self-healing functions and virtualization of capacities that grow as needed? The entire system should be protected. Fortunately, the logical architectures for Technology Foundation directly address complete redundancy.

**Note:** Actually, some fault tolerance is better than no fault tolerance. Moreover, consider the probability of failure when applying fault tolerance to segments of the system. For example, a hardware firewall may be the weak link in a system’s redundancy, but that hardware component may have a far superior mean time between failure rate than PCs in the system, for example. The principle here is that it’s better to protect the entire system, rather than to protect only a portion of it.

While countless variations and hybrid strategies exist, broadly speaking, there are five types of fault tolerance:

- ▶ No fault tolerance
- ▶ High availability (defined in “High availability verses continuous availability” on page 16)
- ▶ Continuous availability
- ▶ Interrupted disaster recovery
- ▶ Uninterrupted disaster recovery

No fault tolerance is the most common option for companies that have not yet experienced or calculated the cost of downtime. This strategy is really only appropriate for pilot programs or proof of concept initiatives.

Both interrupted and uninterrupted disaster recovery options are frequently required. A description of these options is outside the scope of this redbook. However, you may refer to “Disaster recovery” on page 18 for an overview.

#### The business case for high availability

The business case for high availability can be made by comparing the costs associated with high availability to the costs associated with unplanned downtime. The costs associated with unplanned downtime include:

- ▶ Employee idle time  
What is the price incurred to pay idle workers?
- ▶ Lost employee productivity  
How much work would be performed if the system remained operational?
- ▶ Dropped customer in-flight transactions  
How many customers would place additional orders?

- ▶ Denied service for new customer transactions

How many new transactions will be denied because of downtime?

- ▶ Decreased customer perception

Will consumers of the system be forced to subvert the normal business process because of an outage? Will new customers be driven to competing services because of downtime?

The costs associated with high availability include:

- ▶ Cost of additional hardware

With Technology Foundation, additional software procurement costs are not incurred.

- ▶ Recurring cost of software maintenance

- ▶ Performance loss from fragmented resources

Fragmenting Technology Foundation resources frequently results in performance gains.

- ▶ Increased installation costs

Careful thought and analysis are required to answer each of these questions with a dollar amount. The cost, for example, of decreased customer perception is quite speculative. Nevertheless, it may be reasonable to suppose that one half of one percent of all customers will be outraged by an outage and discontinue service. These kinds of assumptions, which are each client-specific, help to convert the intangible costs of, in this example, customer perception into the number of real dollars lost.

It is important to also include all of the hidden costs. For example, when answering the question “What is the price to pay idle workers?”, you may be tempted to multiply the hourly rate of workers pay by the number of employees that are idle to derive a rate of non-productive pay. However, is the salary per hour figure fully burdened? Does it consider vacation pay, sick pay, the employer’s social security contribution, medical benefits, retirement benefits, the cost of office space, electricity, etc.? Obviously it is essential to use a fully burdened rate in the calculation and to include all hidden costs. This rate will be substantially higher than the base hourly rate and reflects the true cost of downtime.

The costs associated with downtimes are often difficult to quantify but are very real. It is a straightforward exercise to compare the two costs associated with both of the previous lists to determine if it makes sense to have a failover capability. Quantifying the values in the first list can be particularly enlightening. However, the exercise is, to some degree, fruitless.

For enterprise applications, business cases that compare high availability against forgoing fault tolerance entirely always show that the cost of high availability is justified. Enterprise applications simply serve too many people and have too many indirect costs associated with their downtime. However, it is more interesting to compare high availability against continuous availability, which is the topic of the following section.

## **High availability versus continuous availability**

For enterprise applications, high availability is mandatory. Although outages are unplanned and unwanted, they are certain to happen sooner or later. If the non-functional capabilities of the application are not supported, then critical business requirements are in jeopardy. But what justifies the additional cost of continuous availability over simple high availability?

The business case for continuous availability is made in a Technology Foundation environment by determining whether 7x24 hours of service is a requirement. That is, you must compare the cost of downtime during an upgrade to the increased cost of continuous availability over high availability.

The costs associated with unplanned downtime are cited in the previous section. The costs associated with upgrading from high availability to continuous availability include:

- Cost of additional hardware

With Technology Foundations, additional software procurement costs are not incurred.

- Increases to the recurring cost of software maintenance

This cost is minor because the alternative, high availability, already incurs the majority of maintenance costs. Continuous availability simply requires operations performed on the backup machines to be executed one more time.

- Increases to installation costs

This cost is minor because continuous availability requires at least three installations rather than two. Economies of scale are expected.

While continuous availability incurs additional costs, it normally comes with some performance benefits. For degraded availability strategies, the system's workload is distributed across the additional machines that accompany continuous availability. In a high availability strategy, two machines perform the work, while with continuous availability, three machines are used.

The sizing process that follows the logical architecture selection process can, at the client's preference, trade the performance benefit associated with continuous availability with cost savings instead. By using three less expensive machines to implement continuous availability rather than two more expensive and capable machines to implement high availability, cost decreases, continuous availability is achieved, and performance remains constant.

### 3.4.2 Determining security requirements

Evaluating how secure a given architecture can only be done by first understanding the relative threat. Many companies have virtually no internal security. They believe that if they are protected from Internet attacks, then they are safe.

While it is true that a *denial of service attack* is unlikely to originate from within the corporate network, nevertheless the majority of security compromises are internal and come from within the Internet firewall boundaries. The majority of security compromises originates from a non-malicious but internal source.

For example, e-mail messages often contain viruses and cause attacks to originate from within firewall boundaries. An employee who commutes with a mobile computer or brings in media may inadvertently corrupt the internal network. For these reasons, critical internal systems, such as enterprise applications, must be protected from the internal network in the same way that they are from the Internet. Internal firewalls are essential. The security characteristics of the Technology Foundation logical architectures are based on varying degrees of this principle.

Security can be implemented to an arbitrary number of degrees. Table 3-3 lists five levels of security although only two of them, the shaded rows, are appropriate for typical EnterpriseOne deployments.

Costs associated with security tend to be more significant than costs associated with high availability. This is primarily because security mechanisms require a greater degree of expertise to implement them. A trained expert in security is essential because configuration mistakes simply cannot be made, particularly if sensitive corporate or customer information is stored within the enterprise applications.

Table 3-3 Levels of security

Level	Characteristics	Appropriate environment
Minimal	No security	Single user environments
Basic	Single firewall and network address translation	Home use or business equivalent
Standard	HTTP servers in DMZ. Firewall configured with Network Address Translation (NAT).	Appropriate for businesses with no Web accessible, sensitive information such as credit card numbers, confidential medical records, financial information, or other proprietary information.
High	DMZ has reverse proxy servers only. Firewall configured with NAT.	Appropriate for a business with sensitive client information. Also appropriate for any business that serves any information over the public Internet.
Extreme	Airlock. No network contact, but protected by all security layers anyway.	Appropriate for national security or environments where total network isolation is essential (beyond the scope of this redbook).

### Standard verses highly secure

The standard security architecture has the advantage of requiring less hardware and software. It is simpler since fewer software components are involved. However, the high security model affords greater protection and compartmentalizes all front-end Web components. Technology Foundation customers need high security (as opposed to standard security) if the answer to any of the following questions is *yes*:

- ▶ Does your EnterpriseOne system or HTTP servers serve information over the public Internet?
- ▶ Is your EnterpriseOne system used to manage client sensitive information, such as credit card numbers, medical information, financial data, or proprietary internal information, where at least moderate damage to the business would be incurred if compromised?
- ▶ Do you have a corporate standard of selecting the most secure system configurations?
- ▶ Do your HTTP servers serve confidential *static* content?

For most clients, the answer to this question is *no*. Static Web content usually consists of simple images of small items that appear on Web pages, such as buttons, or larger images, such as the company logo. They may also serve static text that appears on Web pages identically for all users. Dynamic text content is usually confidential. It is generated by WebSphere and not considered static content. For example, dynamic text that shows a customer's account balance, which varies from customer to customer, is confidential.

If the answers to all of these questions is *no*, you can select *standard security* architectures. Otherwise, you must use *highly secure* architectures. WebSphere security experts occasionally cite the difficulty in managing static content across a firewall as another reason to prefer the highly secure architecture over the standard security architecture. However, this objection is relatively minor.

### Determining your LDAP requirements

Determining your LDAP requirements is straightforward. Consider these questions:

- ▶ Do you have an existing LDAP?
- ▶ Do you want to use it as your authentication source?
- ▶ Is there a supported interface to the external LDAP?

If the answers to these questions are *yes*, you must customize the standard architectures with an external LDAP. Refer to 3.3, "Allowed customizations to logical architectures" on page 29, which discusses LDAP customizations.

## Debunking the myths

The following sections highlight a few key myths about security that permeate the industry.

### ***Myth: If the system is on an internal trusted network it is secure***

Again, the vast majority of security compromises occur from within corporate networks. If your company has experienced a virus, it is far more likely that the virus came from a trusted source, such as an e-mail message, a consultant's PC, or a mobile computer brought in from home, rather than from an Internet source that directly burrowed in through the firewall.

Typically, viruses and worms do not enter through the back door. More frequently they are invited in the front door. Enterprise applications reside on machines that must be far more protected than the average company machines because the cost of problems on those machines greatly exceeds the cost of problems on normal corporate machines.

### ***Myth: If your employees use an encrypted virtual private network (VPN), or a dial-up connection, then you don't have to worry about viruses***

This is a variant of the first myth. It's true that hackers are unlikely to break the encryption codes used for VPNs. It is also true that they don't have to.

### ***Myth: Building security is sufficient to physically protect enterprise application servers***

Do not underestimate the threat of having physical access to machines, especially PCs. All enterprise application servers must be physically secured. The enterprise application network must also be physically secured. Place the physical machines and the network that resides behind the firewall in a restricted access area. Log the entry and exits to that area, so that if a breach occurs, there is a limited group of people with access.

**Note:** PCs are particularly vulnerable. Physical access should be considered universal access. For example, there is a well-known technique to construct a bootable Linux diskette that can change administrator passwords on other well-known operating systems installed on the hard drive.

### ***Myth: A security specialist is not needed to configure security components***

It is extraordinarily risky to rely on the skills of a mere network administrator, particularly if they do not deal with security configuration and installation on a day-to-day basis. Network administrators frequently don't understand how to segment IP address spaces and how to simultaneously provide security and plan for network growth. They often inadvertently open holes in the network when configuring firewall rule trees with the inevitable exceptions. They may not understand the harm of allowing ping traffic or may perfectly configure the firewall rule tree, but then forget to secure the firewall itself, which can also be hacked. Administrators may appear to get WebSEAL to work, but forget to establish exclusive access between the HTTP server and the reverse proxy.

Nearly any competent network administrator can configure the network so it appears to work on the surface and seems secure. However, the actual strength of the full spectrum of the system's security cannot be tested by a non-expert. For these reasons and many others, a security specialist is worth the investment.

Archived

# After installation

After you establish the physical architecture, order the hardware, and install the software, you must properly configure, maintain, upgrade, and possibly extend Technology Foundation. This part covers these post-installation topics:

- ▶ **Configuration:** You can alter several configuration parameters, particularly within WebSphere. However, only a handful of these parameters have a significant effect for back-end server constrained systems such as EnterpriseOne.
- ▶ **Caring for the system:** Technology Foundation maintenance and management techniques are straightforward but often overlooked. This part discusses monitoring techniques for each part of the system.
- ▶ **Extending Technology Foundation:** Finally, this part shows a few possibilities for extending Technology Foundation to better integrate with external components. It covers adding custom portlets, integrating applications with credential vault or MQ, or supporting Internet access.

Archived



## Configuring and tuning Technology Foundation for high transaction volumes

WebSphere and all Java 2 Platform, Enterprise Edition (J2EE) products tend to offer a myriad of tuning parameters that govern all sorts of behaviors. However, for most deployments, only a handful of the parameters cause noticeable performance or execution improvements. This chapter attempts to identify those few parameters that significantly impact performance and quality of service for EnterpriseOne.

In general, parameters that minimize disk access are more important than parameters that optimize memory operations. You must remember that disk input/output (I/O) is measured in milliseconds, while memory operations are measured in nanoseconds.

WebSphere administrators can spend far too much time optimizing, for example, thread pool allocation or garbage collection. It is more effective to ensure that database tables have the correct indexes (EnterpriseOne has reporting mechanisms to check for the existence of the proper database indexes), to check the size of buffer pools, and to validate the execution plans of common Structured Query Language (SQL) statements.

All database access occurs on the EnterpriseOne server. A discussion of the tuning for that server is outside the scope of this IBM Redbook. While this chapter focuses on tuning for Technology Foundation, administrators should never lose sight of this fundamental tuning concept.

## 4.1 pSeries benchmark

The material for this section was originally produced as an output of a Oracle and IBM benchmark test that ran as many EnterpriseOne users as possible against Technology Foundation Version 4 on AIX 5.2 with EnterpriseOne Version 8 Service Pack 2. The standard JD Edwards EnterpriseOne LoadRunner test suite was used.

The benchmark project was divided into two parts:

- ▶ A front-end initiative to configure and debug the Web components and the EnterpriseOne server
- ▶ A separate back-end initiative to *horizontally scale* and stress the EnterpriseOne database

The horizontal scaling effort was intended to tune the database for performance and to identify and address constraints that limit the number of users.

This section discusses the output of the front-end initiative—the Technology Foundation tuning. Tuning WebSphere for EnterpriseOne is different than tuning WebSphere for typical high volume front-end systems. EnterpriseOne transactions have a unique nature. Specifically, the transactions in the EnterpriseOne benchmark suite tests are characterized by the following properties:

- ▶ EnterpriseOne transactions are back-end constrained.

Transactions spend the majority of time waiting for the EnterpriseOne server and the database server. Nearly all EnterpriseOne transactions that make it past the Hypertext Transfer Protocol (HTTP) server to WebSphere go to the back end.

- ▶ Benchmark transactions have a great deal of think time.

Any single simulated user spends most of their time in the automated test tool's think time. A test that runs only a single user still takes over 30 minutes to run.

- ▶ EnterpriseOne Web pages have relatively few HTML, GIF, and other static data transactions.

The HTTP server handles a few requests for static images, but the vast majority of time for any given transaction is consumed by back-end hits. The HTTP server daemons spend most of their time simply waiting on a response or maintaining a keepalive connection.

- ▶ Benchmark load achieves a steady state.

There are no peak transaction times during the test after the initial application ramp up. Ramp speed is the speed that new users are introduced in a simulated run.

- ▶ Benchmark transactions are not managed with traditional WebSphere workload management.

Workload management is explicitly disabled since users are, in effect, assigned to a particular Web application server (Java Virtual Machine (JVM™)) for the duration of the test by the automated testing tool.

**Note:** The recommended settings cited in this section are not appropriate for all production systems. They are specific to high volume EnterpriseOne AIX installations.

Rather than citing parameters that seem to have worked for a test or two, this section attempts to give you a genuine understanding of what the parameters do so that you can make an informed tuning decision.

### 4.1.1 Physical architecture

Figure 4-1 shows the physical architecture used to debug and stress the Web components and EnterpriseOne server.

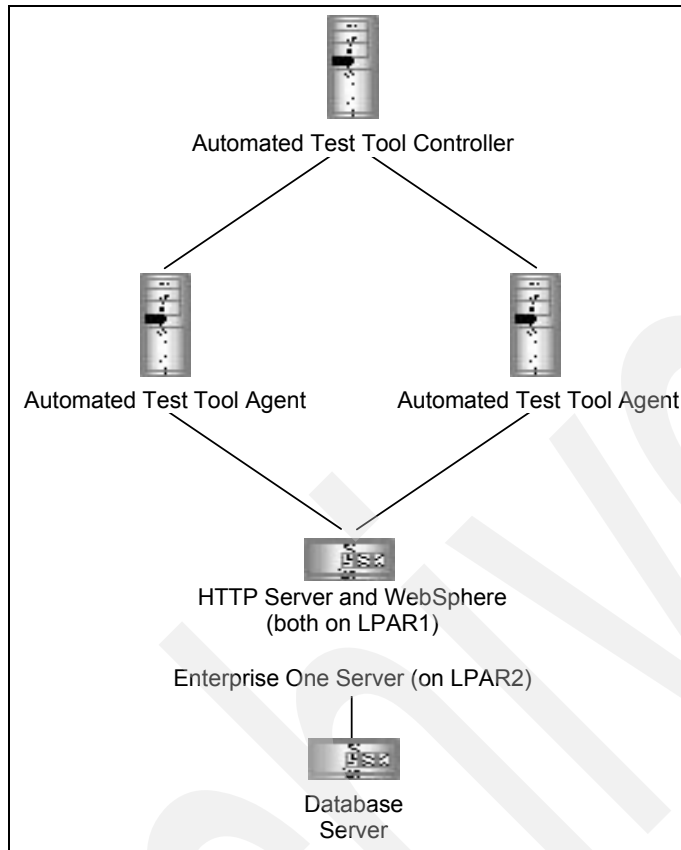


Figure 4-1 Physical architecture for benchmark

The first logical partition (LPAR1) and the second logical partition (LPAR2) were run on the same physical pSeries server. This simple topology was designed to be easily replicated to form a horizontally-scaled architecture, as illustrated in Figure 4-2.

This topology's needless replication and lack of security makes it inappropriate for a production system. For example, it is not necessary to have HTTP servers across so many machines. Network Dispatchers and firewalls are also omitted. Clustering was not used.

This topology was designed solely for benchmarking. It is easily scaled with minimal system administration. This simple yet highly redundant architecture was selected to ensure that the horizontal scaling portion of the project would produce database tuning problems only, rather than creating HTTP, WebSphere, or EnterpriseOne tuning problems. Recall that Technology Foundation tuning was performed separately from the database tuning initiative.

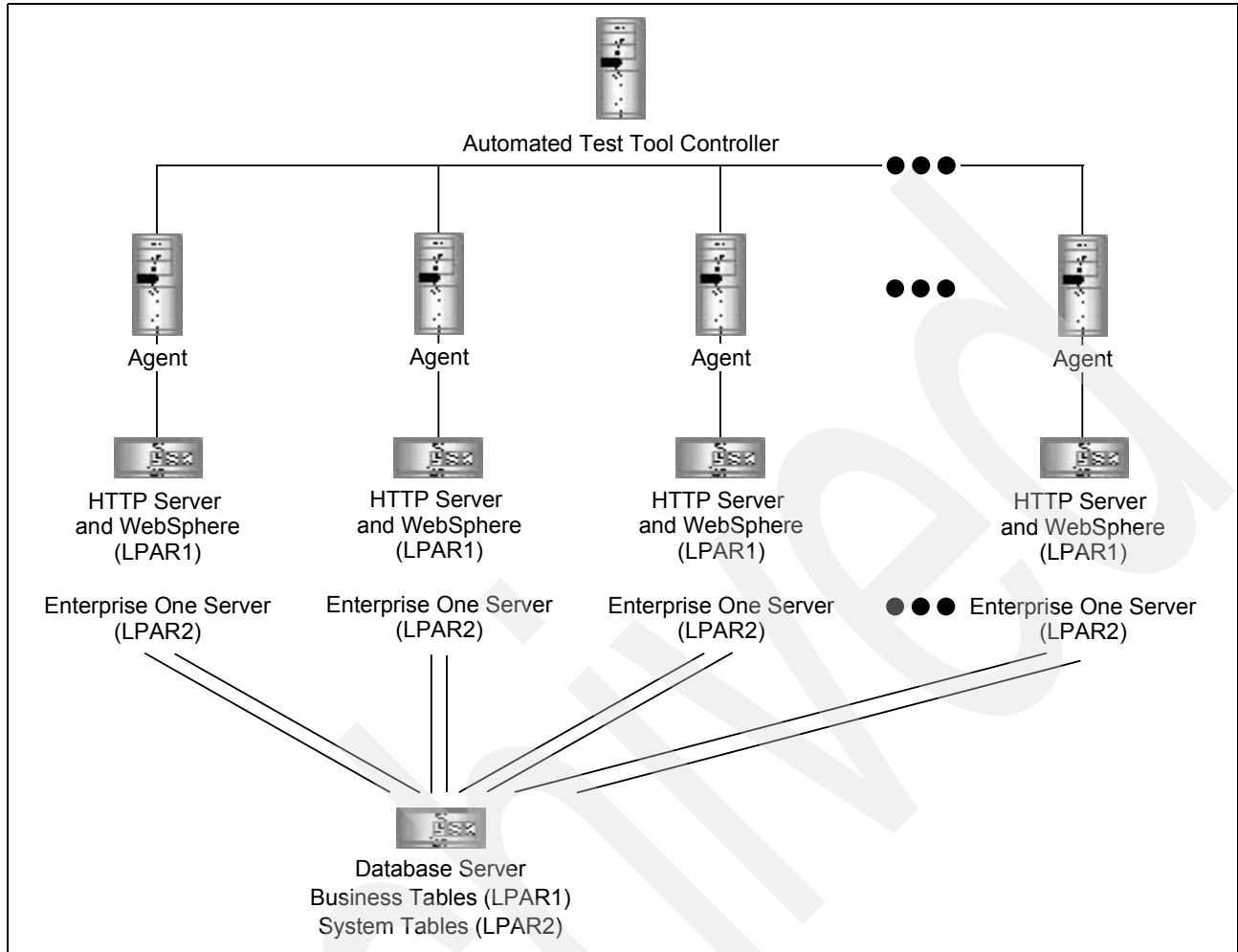


Figure 4-2 Physical architecture for horizontal scaling portion of benchmark

Table 4-1 and Table 4-2 show the hardware configurations for the pSeries benchmark.

Table 4-1 Hardware for HTTP, WebSphere, and EnterpriseOne servers

Hardware component	HTTP, WebSphere, EnterpriseOne servers
Machine	pSeries Model 650
CPU	8
Speed	1.45 GHz
Memory	32 GB (9 GB for LPAR1, 23 GB LPAR2)

Table 4-2 Database servers

Hardware component	Database server for Web component configuration effort (Denver)	Database server for back-end horizontal scaling effort (Beaverton)
Machine	pSeries Model 650	pSeries Model 690
CPU	8	16 (LPAR1 - system tables - 4 CPUs; LPAR 2 - business tables - 12 CPUs)
Speed	1.45 GHz	1.3 GHz
Memory	128 GB	128 GB (LPAR1 - system tables - 28 GB; LPAR2 - business tables - 92 GB)

### 4.1.2 Detailed logical architecture

This section discusses the logical architecture for the benchmark environment that was used to determine peak system load. To identify where bottlenecks occur in a distributed system, it is useful to understand the logical architecture of the system. Figure 4-3 shows this detailed architecture.

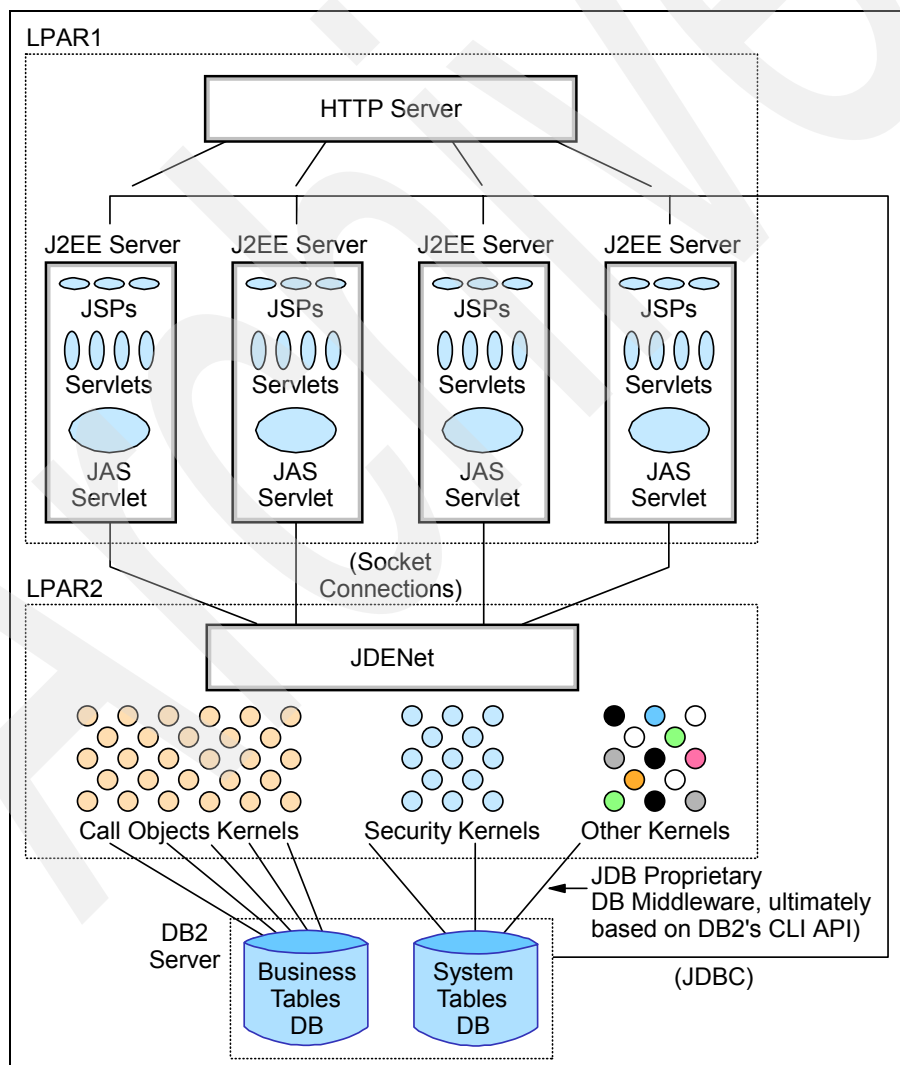


Figure 4-3 Detailed logical architecture

The database can receive requests from two sources: from the JAS servlets (also known as the *EnterpriseOne Web Server component*) or from the EnterpriseOne kernels. The majority of database requests arrive from the EnterpriseOne kernels and are directed against the business tables. The JAS servlets also request information from the database, but this is primarily during the initial login sequence.

JAS goes directly to the database to authenticate and cache user interface information. It also gets data to determine how to map certain users groups to certain tables. For example, the next numbers tables (tables that assign unique numbers) can be mapped to different user sets.

The system tables are read-only during normal operation and are written to only during configuration scenarios (not encountered in the benchmark scripts). The system tables contain configuration settings, security information, custom graphical user interface (GUI) information, and information about mapping individual users to different database tables to distribute load.

**Note:** The custom GUI information is informally referred to by JD Edwards EnterpriseOne as *serialized objects* because of the nature of their particular Java implementation.

No complex interactions between the two sets of database tables occur, and no two-phased commits are ever needed to coordinate commits. The business tables database is the place to focus the majority of optimization efforts, although is out of scope of this document.

For the Technology Foundation configuration portion of the project, a single database was used to serve both the business tables and the system tables. For the horizontal scaling portion of the project, these two sets of tables were served by separate database servers.

Communication from the JAS servlets to the kernels is through the JDENet layer, which brokers each transaction. Each kernel is its own UNIX process, and tools exist to monitor their work queues. See 5.2, “Operations monitoring and management tools” on page 71, for a further discussion.

While there are several different kernel types, the most noteworthy type is the *call object kernels*, which do the vast majority of work. The call object kernels are currently singly-threaded C and C++ processes that queue up work internally. The call object kernels cache large amounts of information for the users they service. Therefore, as users submit new requests, it is important that they are mapped to the same kernels that processed their previous requests. Because of this affinity, session-level failover in WebSphere is not used.

## 4.2 Recommended parameter values and reasoning

The parameters that are appropriate to tune depend on the tuning objectives. For example, the HTTP MaxClients parameter should be raised if a large number of users is going to be supported. However, that same parameter has little effect on response time. Conversely, altering the heap size has more to do with improving the performance for a single user, but does not have a profound effect on increasing the systems ability to support high transaction volumes.

If any tuning parameter is misconfigured, it can have a profound effect on both transaction volumes and response time. If heap size is set too low it can affect both the response time and the system's ability to support large volumes of users. Nevertheless, in this redbook, the parameters are categorized based on the tuning objective that is best served as the given parameter is increasingly altered from its default value. The following sections present

parameters that primarily help to prevent transactions from being dropped and improve performance.

#### 4.2.1 Tuning parameters that prevent dropped transactions at high volumes

The tuning parameters in the following sections are critical for systems that are expected to serve high transaction volumes and that extensively hit back-end components. The parameters described in this section prevent the system from simply dropping a user's transaction during heavy load.

EnterpriseOne systems primarily stress the back-end system components. Business environments that stress front-end components, such as a Web site that serves huge quantities of static data, emphasize completely different parameters to support high transaction volumes since serving static data is the job of the HTTP server and not WebSphere (if properly configured).

Moreover, administrators who configure systems that are not subject to high transaction volumes (even EnterpriseOne systems), but are instead more concerned with response times for a small number of users, may elect to skip this section and focus on "Parameters that enhance response time" on page 59. The parameters in this section, if misconfigured, tend to cause dropped transactions rather than poor response times for high volume systems bottlenecked on back-end components.

##### HTTP

In general, the default IBM HTTP Server parameter settings require no adjustment even for relatively high transaction volume EnterpriseOne systems. HTTP daemons are lightweight and lightning fast.

IBM HTTP Server (powered by Apache) is a repackaging of the Apache Web server, which is extraordinarily well tested. Apache is used to serve more than half of all Web pages on the Internet. It is generally the wrong place to look for bugs. While HTTP server configuration changes can be necessary for systems with extremely high numbers of HTTP transactions, in the case of EnterpriseOne transactions, this is rarely, if ever, the case. In an EnterpriseOne environment, the time consumed in the HTTP server is dwarfed by the time spent waiting on the back-end. Virtually every transaction goes to the back end. Consequently, the HTTP server is extremely unlikely to be a bottleneck.

##### *MaxClients*

MaxClients is the only HTTP server parameter that qualifies as critical since, if set too low, results in immediate dropped transactions at the front end. If Apache reaches the limit of child processes, it immediately responds back to the end user with an error. In the case of a benchmark, the automated testing tool stops making requests on behalf of that virtual user.

The default value of 1024 did not require alteration in this project's benchmark. When running the standard EnterpriseOne benchmark test scripts on a single AIX server for 2000 users, the number of HTTP daemons hovered around 600 processes. The number of HTTP daemons can be counted with the UNIX command `ps -efl | grep -c httpd`. Benchmark tests were unable to reach the 1024 limit unless requests were simply submitted faster than the back end could work them off (in which case any number of daemons is eventually consumed). Even at 10,000 users, the limit was not reached.

Nevertheless it is conceivable that, when processing extremely high bursts of EnterpriseOne transactions, this value may need to be increased. The value should be increased to the maximum number of concurrent connections under peak load. Since any single Web page

can establish multiple connections to the HTTP server, avoid confusing the number of connections with the number of users or the number of concurrently hit Web pages.

A ratio of five connections per Web page is reasonable, but calculating an appropriate parameter value is most easily done empirically. If the limit is reached, in the Apache error\_log file, you see the error message: MaxClients limit reached – consider increasing this value.

A common mistake is to confuse reaching this limit with the limit itself being the source of the problem. More often, the limit that is being reached shows there is a problem on the back end causing it to run too slowly. If the back end runs too slowly, this limit is reached regardless of how high it is set. This error message was encountered several times during the course of the benchmark project, but never during a normal run, although no peak bursts were simulated during the benchmark test. As back-end problems and configuration settings were fixed, MaxClient violations always disappeared.

The cost of setting the value higher than the already high default is that the operating system may attempt to spawn more processes than it has the capacity to support. This cost is minimal. Setting the number to higher values does not automatically cause new HTTP daemons to be created. They are only created if needed and this growth is controlled by the minimum and maximum spare servers parameters (see “Tempting but insignificant parameters” on page 63). However, if this value is set high and if the back end is bottlenecked, the HTTP server may attempt to spawn more processes than the operating system can support. This may cause additional, severe problems with any other processes running on the HTTP server machine.

It is important to re-emphasize here that this section is only about AIX and not Windows or iSeries. The Apache MaxClients implementation on Windows works differently than on AIX, for example. In fact, the MaxClients parameter does not exist for a Windows deployment.

**Recommended initial value:** MaxClients = 1024 (defaults to 1024)

## **WebSphere Application Server parameters**

Consider altering the following WebSphere Application Server parameters if you encounter dropped transactions during heavy system load.

### ***Max thread size and allow thread allocation beyond maximum***

While the minimum thread size parameter’s effect on performance is imperceivable, running out of threads is a severe problem and causes transactions to be dropped. The documentation for how these two parameters are used internally by the WebSphere thread pool management is somewhat deficient, but is clarified in the following note box. “WebSphere thread allocation” on page 55 explains the algorithm that WebSphere uses to manage the thread pool.



**WebSphere thread allocation algorithm clarification:** When WebSphere Application Server starts, it immediately creates the minimum number of threads. The number of threads in the thread pool never drops below this amount in the lifetime of the JVM.

If more threads than the minimum number are needed as WebSphere Application Server runs, it creates them. If Allow thread allocation beyond max is not selected, WebSphere does not create more than the maximum number of threads, even if it needs more.

If a thread remains idle (a thread waiting for a back-end response is not considered idle) for more than the “thread inactivity timeout”, then it is destroyed.

If WebSphere Application Server needs more threads, but it cannot create them, it does not respond to the HTTP server’s next request. A lack of response signals the HTTP server to mark the application server as *down* (visible in the native.log file) and use by default a round-robin algorithm to select a less busy application server.

If WebSphere Application Server needs more threads and Allow thread allocation beyond max is selected, it creates more threads than the maximum number. However, when the thread is freed up, it deletes the thread so that the number of threads returns to the Maximum number of threads parameter.

### **WebSphere thread allocation**

In heavily loaded systems, application servers with the Allow thread allocation beyond maximum parameter as unavailable show a total number of threads that hovers just under the maximum allowed number. Occasional drop outs are observed. However, if the Allow thread allocation beyond maximum parameter is available, the total number of threads hovers just over the maximum allowed number since new threads are constantly created and destroyed. This behavior is easily observed with WebSphere Resource Analyzer.

Many WebSphere tuning documents recommend the minimum thread pool size to be set to the same value as the maximum thread pool size. This is the recommended best practice for several reasons:

- ▶ The performance benefit realized from dynamically creating and destroying threads is negligible for the vast majority of systems. Allocating or destroying a thread is an in-memory operation and, as such, is very fast, especially compared to disk I/O operations.
- ▶ If the system has to dynamically allocate resources for a thread, administrators may not learn about this until peak load time that the physical machine cannot handle the maximum number of threads. The JVM can become marked down during production.
- ▶ In older versions of WebSphere, several problems were associated with the dynamic creation and destruction of threads. Setting the parameters to the same value in these older versions avoided bugs associated with dynamically managing the number of threads in the pool.

To see the problems with JVMs being marked as down, it is essential to closely monitor the native.log file. The location of the native.log file is specified in the /usr/WebSphere/AppServer/config/plugin-cfg.xml file. By default, it is set to /usr/WebSphere/AppServer/logs.

This file is not owned by WebSphere. Administrators frequently clear WebSphere’s log file prior to starting the administrative server, by entering the following UNIX command, which is ordinarily a defensible practice:

```
rm /usr/WebSphere/AppServer/logs/*
```

However, since the native.log file is actually owned by the HTTP server, this operation can accidentally delete an active log file.

If the native.log file is deleted when the HTTP server is up, the server simply stops recording messages to that file. If a JVM is marked down sometime later when the system is under stress, and if the native.log file is deleted, then there is no record of the problem in any log file. Obviously there are several solutions to this problem. However, the essential concept here is to use care to ensure that the native.log file is not deleted when the HTTP server is up.

#### **Recommended initial values:**

- ▶ Min Thread Size: 100 (defaults to 10)
- ▶ Max Thread Size: 250 (defaults to 50)
- ▶ Allow Thread Allocation Beyond Maximum: Yes (default is No)

### **Keepalives**

After several experiments with both small and large test runs, the following values were set to zero:

- ▶ Maximum keepalives = 0
- ▶ Maximum requests per keepalive = 0
- ▶ Keepalive timeout = 0

These values effectively turned off WebSphere keepalives. This is not to be confused with HTTP keepalives, which remain turned on.

**Note:** The keepalive entry in the glossary describes the two types of keepalive connections found in WebSphere. Make sure that you understand the difference between these two when altering configuration settings.

These recommended settings are certainly not consistent with traditional tuning advice that recommends either setting the keepalive values relative to a certain percentage of the number of threads or setting the values relative to the number of waiting socket connections reported by the netstat utility.

**Note:** Consult the Technote *TCP/IP sockets in CLOSE\_WAIT state on a Web server loaded with either the WebSphere Application Server V4.0, V5.0 or V5.1 plug-in module* to learn about the technique of using netstat to determine appropriate keepalive values. You can find this Technote on the Web at:

<http://www.ibm.com/support/docview.wss?uid=swg21163659>

This technique is a defensible practice if you do not encounter the problems that are described in this section and associated with having keepalives turned on. However, for back-end bottlenecked systems, the keepalive values have little or no noticeable effect on performance (at least for 14,000 users). Turning them off entirely is easier and safer.

To understand these unconventional recommendations, it is important to understand that WebSphere (and HTTP) keepalive timeout values govern the behavior of a connection after the response is received and the connection is ordinarily closed. Keepalive parameter settings alter the behavior of idle connections and not active connections. Therefore, keepalive values have no effect on active transactions that are blocked waiting for the EnterpriseOne server since that transaction is considered “in process”. This is true for any back-end bottlenecked application. It is not EnterpriseOne specific.

You can easily confirm these statements by configuring a low keepalive timeout value and running a servlet that sleeps for several minutes. The connection to the sleeping servlet is not lost. Keepalives are simply used to force a TCP connection, which is otherwise closed, to remain open to avoid the overhead associated with re-establishing the connection during a series of transactions between the same client and server.

You can safely turn off keepalive optimization between the HTTP server and WebSphere for EnterpriseOne systems.

**Note:** This is true even if Secure Sockets Layer (SSL) is enabled. SSL is generally viewed as a protocol that makes keepalives particularly important since a separate security transaction is performed every time the connection is used. While this is true, for back-end bottlenecked systems, the additional connection setup time is still insignificant relative to the time spent waiting on the back-end request. SSL itself can cause noticeable degradation of performance, but not due to the associated connection re-establishment.

The time that WebSphere takes to set up a new connection is insignificant compared to the time spent waiting on the back-end components. In such systems that ultimately wait for database transactions, keepalives have little real benefit. However, WebSphere keepalives come with several serious drawbacks, especially on UNIX-based systems. These drawbacks include:

- ▶ UNIX keepalives can tie up threads in some operating systems. Socket and thread behavior is radically different between PCs and UNIX. On the PC, the keepalive connection binds the HTTP server to the Web container. In AIX 5.2 UNIX, WebSphere keepalives bind the HTTP server to a particular thread. As the connection remains idle, so does the thread. This assertion can be proved with the following simple experiment:
  - a. Install an application (any application) in WebSphere.
  - b. Create a new WebSphere Application Server server with all the default settings.
  - c. Configure that WebSphere Application Server server with minimum and maximum threads both set to 1.

**Important:** If you perform this experiment in WebSphere Application Server 5.x, back up your server.xml file. Changing the values back to the original values through the Web-based administrative console may not be possible with only one thread. In WebSphere Application Server 5.x, the keepalive and connection backlogs values are configured as custom properties from the admin console by selecting **Servers → Application Servers → server1 → Web Container → HTTP Transports → [host name] → Custom Properties → New**.

See the following Web site for a list of acceptable custom property names:

<http://www.ibm.com/software/webservers/appserv/doc/v40/ae/infocenter/was/06061300.html>

- d. Configure that same server to have a noticeable keepalive timeout value, for example, 30 seconds.
- e. Save the settings and restart the server.
- f. Open two browsers on two different computers.
- g. Type the URL for the application on both browsers. Then press Enter on both computers at the same time.

Notice that, if your WebSphere Application Server server is running on a PC running either Windows or Linux, both browsers return nearly instantaneously even though there was only one thread to process both requests. However, if your WebSphere Application Server server is running on AIX 5.2, one browser returns immediately, while the other browser waits the exact same number of seconds as the keepalive timeout value.

As threads become tied up, the workload is increased on all remaining threads. Since AIX 5.2 keepalive values force the application servers threads to remain idle, the remaining threads are forced to process more requests. As they process more requests, they become increasingly likely to encounter a user in “think time” and become idle themselves. The rate of the increase of forced idle threads depends on the rate of increase of the workload which, in turn, depends on the rate of increase of forced idle threads. While some dampening factors exist, this circular relationship results in logarithmic decay. It is a recipe for sudden disaster since all threads become idle and JVMs are marked as “down” by the HTTP server.

- Small keepalive timeout parameter values occasionally cut off live connections. Keepalive timeout values set under 10 seconds (but not zero) occasionally cause the application server to snap the connection closed in the middle of communication with the message:

```
(Thu Jul 31 11:01:16 2003] 00006de0 00000001 - ERROR: ws_common: websphereExecute:
Failed to read from a new stream; App Server may have gone down during read
[Thu Jul 31 11:01:16 2003] 00006de0 00000001 - ERROR: ws_server:
serverSetFailoverStatus: Marking AS_JDEdwards_4 down
[Thu Jul 31 11:01:16 2003] 00006de0 00000001 - ERROR: ws_common:
websphereHandleRequest: Failed to execute the transaction to 'AS_JDEdwards_4'; will try
another one
[Thu Jul 31 11:01:16 2003] 00006de0 00000001 - ERROR: ws_common:
websphereHandleRequest: Failed to find an app server to handle this request
```

This is a known bug in WebSphere Application Server 4.x. It is not fixed by the latest cumulative keepalive plug-in patch, which you can find on the Web at:

[http://www.ibm.com/support/docview.wss?rs=180&context=SSEQTP&q=keepalive&uid=swg24001801&loc=en\\_US&cs=utf-8&lang=en](http://www.ibm.com/support/docview.wss?rs=180&context=SSEQTP&q=keepalive&uid=swg24001801&loc=en_US&cs=utf-8&lang=en)

If the tests described in this section demonstrate that your threads are not consumed by keepalive connections or that they are not snapped off prematurely, then you can safely leave the settings at their default values. However, you may want to turn off HTTP server to WebSphere keepalives since the settings don't significantly impact performance of a back-end bottlenecked system. Keepalives can always be turned back on and tested later.

During the pSeries EnterpriseOne benchmark initiative, keepalive and connection backlog parameters were fixed early in the project. While WebSphere was constantly monitored, all subsequent advancements that increased the number of users were related to database tuning. The keepalive and backlog values freed the system from its 3,600 user limitation to run a sustained load well beyond 10,000 users.

#### **Recommended initial values:**

- Maximum Keepalives: 0 (defaults to 25)
- Maximum Requests Per Keepalive: 0 (defaults to 100)
- Keepalive Timeout: 0 (defaults to 5)

#### ***Connection backlog***

The connection backlog works in conjunction with the HTTP servlets to be the first place requests can be buffered. While the HTTP server buffers requests by creating more HTTP daemons, it only does so if the daemon's connection request to the WebSphere Web

container is not immediately refused. The backlog parameter keeps WebSphere from rejecting new requests from HTTP daemons.

Several WebSphere tuning documents recommend that you keep the system load shifted to the front-end components to avoid overloading the back end. While it is better to buffer requests and process them only when the system can handle the additional load, administrators must keep in mind that, on average and over time, the overall system has to process requests as fast as it receives them. If it fails to do so, all buffers eventually fill up and overflow, regardless of how large they are or where they are. The technique of buffering transactions is only good for smoothing out transaction bursts and to help the system achieve a steady workload.

The connection backlog parameter's documentation is somewhat deficient, but is clarified here. To understand how the call works, it is helpful to know that WebSphere ultimately passes the parameter to the constructor for the `java.net.ServerSocket` class. One of the parameters to that constructor is the backlog parameter. Therefore it is possible to understand the parameter better by examining the Javadoc™ for that call. In the Javadoc, the backlog parameter is described as:

“The maximum queue length for incoming connection indications (a request to connect) is set to the backlog parameter. If a connection indication arrives when the queue is full, the connection is refused.”

The connection backlog pertains to buffering requests to instantiate new connections. It does not refer to buffering data transmitted over active connections.

In the benchmark project, we changed the value from a default setting of 10 to a safe value of 8000. When values higher than 8000 were attempted, the JVMs could not start. That is because UNIX kernel resources are preallocated and consumed when the JVM is started, which is probably why the upper limit exists. The value of 8000 was probably excessively high anyway since WebSphere really only needs to buffer as many incoming connection requests as it can possibly ever receive. Nevertheless, this value certainly has been validated for over ten thousand simulated users and is therefore the officially recommended value (there was not time to test lower values). In UNIX, because each HTTP daemon is singly threaded and the total number of daemons is limited by the `MaxClients` parameter, setting the connection backlog slightly higher than the value of `MaxClients` probably is sufficient.

**Recommended initial value:** Connection BackLog: 8000 (defaults to 511)

### EnterpriseOne servlet configuration

The `UserSession` timeout value in the `jas.ini` file was increased from the default value of 1,200,000 milliseconds (20 minutes) to 12,000,000 milliseconds (200 minutes). The value was changed inordinately high to ensure that all errors were ultimately returned to the user interface. When the debugging process for the benchmark initiative was complete and very slow transactions were eliminated, the high value was no longer necessary. However, the high value caused no harm so it remained for the duration of the project. For production systems, the original default value of 1,200,000 is fine.

**Recommended initial value:** UserSession: 12000000 (defaults to 1200000)

## 4.2.2 Parameters that enhance response time

The following parameters increase response times but do not heavily impact an EnterpriseOne deployment's ability to handle additional users. The default values for the parameter settings in this section typically do not cause dropped users.

## HTTP parameters

Given the nature of EnterpriseOne transactions, tuning parameters in the IBM HTTP Server (powered by Apache) do not result in a significant performance improvement. MaxClients, Min and max spare servers, Keepalive values, MaxRequestsPerChild, etc. all have good default values. All of these parameters configure the management of IBM HTTP Server (powered by Apache) in-memory operations performed by extremely lightweight daemons. Unless IBM HTTP Server (powered by Apache) is dropping requests, it does not cause problems (refer to “MaxClients” on page 53). Any of the process management operations are much faster than the fastest database operation. These values are more appropriate for tuning systems with huge quantities of small requests for static data (such as images). By contrast, EnterpriseOne systems have small quantities of long duration back-end transactions.

**Note:** WebSphere tuning advisors often recommend a low MaxClients value. This successfully lowers the load on the back-end components and does so at the cost of denying users access to the system on the front end. Therefore, for a benchmark or high volume production system, low values are typically not an option.

HTTP configuration parameters are particularly platform specific. The Apache-based implementation for UNIX is entirely different than for the PC. PCs use fewer multithreaded processes, while UNIX uses many singly-threaded daemons. Configuration parameters from one platform may be entirely absent on another. When taking advice or copying parameters from other projects, consider the platform. Further, consider whether there is a genuine understanding of the changed values. Fine tuning parameters can be altered to very high or low values with little or no effect on results. For example, changing the HTTP setting MaxClients from 1024 to 1,000,000 is likely to have no effect.

## Number of JVMs in WebSphere Application Server

Is it better to have one JVM with 800 threads or four JVMs each with 200 threads? Is it better to have one JVM with a heap size of 1024 or two JVMs with heap sizes of 512 each? If performance is increased by adding an additional JVM, can you realize the same benefit by simply increasing the heap sizes on the original JVMs? To answer these questions, an administrator must understand what is genuinely and uniquely gained by adding a JVM and how the operating system manages threads.

The following list summarizes the benefits that are realized by adding a JVM that cannot be achieved by simply expanding capacity in existing JVMs:

- ▶ Web application server failover  
Additional JVMs are used as a backup for any single JVM failure.
- ▶ Very large heap size support  
Even by altering the LDR\_CNTRL (described in the following section), a single JVM can support at most 2.5 GB in AIX 4 or 5.
- ▶ Additional transport connection  
Each JVM comes with its own primary port number. However, internally, communication is immediately delegated to another socket connecting the HTTP daemon to a WebSphere Application Server thread, so the magnitude of this benefit is questionable.

The following list summarizes the costs that are realized by adding a JVM and that are not incurred by expanding capacity on the previously existing JVMs:

- ▶ Thread management overhead  
Each JVM must manage context switching between threads.
- ▶ Administration complexity  
The application must be deployed to each JVM or a server group must be created.

Extensively testing the trade-offs between increasing heap size, while simultaneously lowering the number of JVMs (and vice versa), was not performed since greater benefits were likely to be realized by experimenting with other settings. However, a few tests demonstrated that adding more JVMs slightly improved response times. These tests essentially tapped into unused system capacity and did not involve trade-offs between memory and more JVMs.

**Recommended initial value:** Number of JVMs: 6 (1 GB heap each on a 4-way box)

Given that many administrators are uncomfortable with altering the LDR\_CNTRL environment variable, a good rule of thumb is to create JVMs with 1 GB heap sizes each until 80% or so of the machine's physical memory is consumed. See , "Heap size" on page 62.

You must reserve space for the operating system and the process heap. EnterpriseOne servlets do not consume an inordinate amount of process heap. You must also reserve additional memory and CPU for any other applications running on the same machine.

The most common intent behind spreading an application across multiple JVMs is to increase performance. However, simply introducing multiple JVMs without an understanding of the trade-offs involved can decrease performance by introducing the overhead of context switching. In general, use fewer JVMs to minimize overhead. The guidelines developed for the benchmark project are summarized here.

Increase the number of JVMs when:

- ▶ You can run the JVMs on separate machines, thereby achieving true parallelism.
- ▶ A JVM can have its own CPU, thereby achieving partial but significant parallelism.

**Attention:** Use care when applying this guideline to Intel platforms. PC operating systems can be quite large, and a single CPU on a multiple CPU machine should be reserved for the operating system itself. The database needs at least one CPU as well. Significantly diminished returns when scaling beyond four processors is a common observation.

- ▶ JVM memory limitations are a concern. Processes (including JVMs) on many platforms tend to encounter a 2 GB memory limitation. There are ways around this, such as altering the LDR\_CNTRL variable on the pSeries platform. However, these ways are complicated to install and maintain, especially if the system administration expertise is limited at the end-client site.
- ▶ A large number of packets may need to be buffered at the operating system level, perhaps due to large bursts of requests. Multiple JVMs, each listening on separate ports and having their own operating system buffer, helps to increase the total amount of packets that can be buffered.

**Note:** A big buffer is not necessarily a good way to deal with performance problems. Ultimately even big buffers eventually fill up if the system's average arrival rate exceeds the average processing rate.

- ▶ Garbage collection causes regular, noticeable delays. This condition is rarely encountered in EnterpriseOne and Technology Foundation deployments.
- ▶ You want to provide individual JVMs for different applications because of security, monitoring or problem isolation purposes.

Chances are that one or more of these conditions are true and the application should be split across multiple JVMs. However, do not add separate JVMs without a clear understanding of why the addition helps. Increasing the size of existing JVMs is preferable.

## Heap size

In general, little is lost by increasing the heap size as long as the additional memory is not required by other processes running on the same machine. By increasing the heap size, garbage collection is not performed as often (although when it is performed, it takes slightly longer). With 2000 users running against four JVMs each with 768 MB heap, garbage collection was observed to occur once every 10 or 15 seconds. By increasing the heap size to 1024, the average response time of OK button transactions was slightly reduced.

**Recommended initial value:** Heap size: 1024 (1 GB) (defaults to 512 MB)

EnterpriseOne servlets use Java Database Connection (JDBC™) drivers to make DB2 client-side calls. This means that the servlets consume memory outside of the JVM heap. Some memory on the physical machine must be reserved for this use. Increasing the JVM heap naturally consumes memory from the process heap. The benchmark initiative never gave the JVMs more than 6 GB of Java heap on a 9 GB LPAR, leaving plenty of room for process heap and operating system memory needs.

When using a 32-bit JVM in AIX versions 4 and 5, it is not possible to allocate more than 1 GB of memory to a single JVM without altering the shell environment variable LDR\_CNTRL. This is an operating system limitation resulting from the memory segmentation design. For a Korn shell (ksh), the syntax for setting the environment variable is `export LDR_CNTRL=MAXDATA=0xn0000000`, where *n* ranges from 0 to 5.

Use this environment variable with care. Once you set it, it causes all subsequent processes that run from the shell to allocate memory differently. Set this variable before the operating system begins to run the JVM processes. Keep in mind that allocating a Java heap comes at the price of shrinking the remaining process heap. You can learn more about the use of LDR\_CNTRL on the Web at:

<http://www-128.ibm.com/developerworks/eserver/library/es-Javaperf1.html>

## Enabling dynamic caching

WebSphere can cache dynamically generated HTML that was generated from servlets and JavaServer Pages (JSPs). To understand this capability, it is important to understand the difference between static and dynamic Web content.

Examples of static content include such image files as GIF files, JPG files (perhaps of the corporate logo or of the appearance of buttons on the Web page), and sound files. HTML files are also considered *static* if they do not contain any customized information for the particular request. For example, an HTML file that contains the address and phone number of the



company may be static since this information does not change. An HTML file containing a stock quote is considered to have dynamic Web content since the information had to be tailored in response to the specific request of the user. The HTML must contain the price of the specific stock in which the user is interested.

Dynamic generation of HTML can be triggered in response to parameters provided by the user, as well as by the time of day, the geographic location of the browser, or by previous actions or pages viewed by the user. Dynamically constructed HTML incurs far more overhead than static Web content since WebSphere, the Directory Server, Portal, EnterpriseOne, the database and many other back-end systems may be involved to assemble the requested information.

While dynamic Web content is specific to a particular user's request, multiple users frequently ask for the same dynamically generated information. This phenomenon of common interests is what makes caching this dynamically generated information make sense. For example, while a dynamically generated HTML file containing a stock quote is specific to a user's request, it is likely that many users in the company are interested in the same stock, perhaps because it is the company stock or because the stock is significantly affected by a certain event that day. Similarly, multiple users are likely to become interested in the same information at the same time whether it's a stock quote, viewing a particular order, checking shipping status, or checking the weather. By caching dynamic Web content, you can prevent numerous back-end transactions.

WebSphere provides a capability known as *Dynacache*, which helps administrators to mark certain servlets (or JSPs, which are ultimately servlets too) as good candidates for caching. Caching of dynamic information makes increasing sense as the Web content industry progressively evolves away from static content toward customized Web pages where users design their own look and feel. You can learn more about Dynacache in any WebSphere configuration manual after Version 3.5.

### 4.2.3 Tempting but insignificant parameters

The following parameters sound promising but should have no effect on an EnterpriseOne deployment. The parameters are divided into two categories: HTTP and WebSphere parameters.

#### **HTTP parameters**

The HTTP parameters that should have no effect on an EnterpriseOne deployment are Min and max spare servers, Keepalive timeout, and Max keepalive requests.

##### ***Min and max spare servers***

The Min and max spare servers parameters control the growth and destruction of the number of idle HTTP daemons. The HTTP server model is comprised of a primary root process that listens on port 80. As soon as a request comes in and faster than a project manager, the root process pawns off the work to a child daemon. The child daemon arranges to meet the client on another socket freeing up the root's port 80 for the next incoming request.

For an EnterpriseOne deployment, the amount of time spent in a transaction dwarfs the amount of time needed to create additional HTTP daemons. Consequently, response times are unlikely to change regardless of the values to which these parameters are set. The default values are fine.

##### ***Keepalive timeout***

This keepalive timeout value governs the behavior of socket reuse for connections between the user's browser and the HTTP child processes. It has nothing to do with the keepalive

timeout between the HTTP daemons and WebSphere. Keepalives dramatically improve performance for sites that have a large volume of short lived requests. While it's true that each EnterpriseOne Web page serves several small images, almost every page is accompanied by a back-end request that takes much longer than the accompanying short requests. Consequently, the HTTP daemons can easily handle the workload of the small requests without keepalives. EnterpriseOne installations are unlikely to realize any noticeable benefit from changing the default HTTP keepalive settings.

By default, all HTTP transactions from the browser to the HTTP server are keepalive transactions. This can be confirmed by turning on the HTTP server access.log file and adding a "%c" to the line that controls the information that is written to the log for each request.

As with the Min and max spare servers parameters, the user is unlikely to notice any difference at all by altering this parameter. For an EnterpriseOne deployment, the amount of time spent in a transaction is greater than the amount of time needed to create a socket connection. The default value is sufficient.

### ***Max keepalive requests***

This keepalive-related parameter governs how many times a connection can be reused. As with the other parameters in this section, the user is unlikely to notice any difference by altering this parameter. Again, the amount of time spent in an EnterpriseOne transaction far exceeds the amount of time needed to create a new connection.

## **WebSphere parameters**

The WebSphere parameters that should have no effect on an EnterpriseOne deployment are discussed in the following sections.

### ***Any parameter pertaining to Enterprise JavaBeans™***

Since EnterpriseOne does not use the Enterprise JavaBean (EJB)-related capabilities of WebSphere, EJB-related parameter settings have no effect. This eliminates several configuration displays. Changing any parameter pertaining to EJB Containers or with the Object Request Broker (ORB) has no consequence.

### ***Maximum in-memory session count***

Since EnterpriseOne deployments tend to use only the in-memory session state management capabilities in WebSphere, this parameter has little effect on either dropped transactions or performance. The in-memory session count is a complicated parameter. It means different things depending on how other session management configurations are set.

The parameter sounds like a limit, but it certainly does not behave like one. If the limit is passed, nothing is actually dropped. If the limit is passed, the sessions are not managed as effectively internally as when they stay below the limit. Furthermore, if the limit is passed and The Allow overrides option is not selected, still no transaction is dropped. In this case, a *dummy session* is created to manage the session.

Since EnterpriseOne installations typically do not use persistent sessions, all of these session management algorithms are in-memory operations so their performance differences are not noticeable. Regardless of how this parameter was changed in sample test runs, problems or noticeable performance degradations cannot be induced.

### ***I/O timeout***

Since the I/O timeout parameter's documentation is not specific as to the kind of connections this parameter affects, you must know how WebSphere uses this parameter internally. According to WebSphere Development, this call is ultimately used as a parameter to `java.net.socket.setSoTimeout(int timeout)`. Sun's javadoc for this routine states:

“With this option set to a non-zero timeout, a read() call on the InputStream associated with this socket will block for only this amount of time.... A timeout of zero is interpreted as an infinite timeout.”

Therefore, the I/O timeout parameter controls how WebSphere treats active connections. It does not affect how idle connections are treated. When WebSphere begins to read data from a socket, the read must finish in 5 seconds (the default parameters value). In EnterpriseOne deployments, a thread is unlikely to transfer so much data through the socket connection from an HTTP daemon that the data transfer will ever take 5 seconds.

### ***Servlet caching***

Experiments with altering the servlet caching values were not performed on the benchmark project since the changes were deemed to have a low probability for a significant effect. Servlet caching is most effective when users often use the same specific data, such as a simple stock quote, where the security symbol is user specific, but common to several different users. Most EnterpriseOne servlets encounter highly specific data.

### ***Thread inactivity***

While the online help describes this parameter setting as “the period of time after which a thread should be reclaimed due to inactivity”, a transaction that is bottlenecked on the back end is not considered inactive. Only threads with no pending transaction are deleted and only if the deletion does not cause the thread count to fall below the minimum thread size.

Threads that are over the maximum thread size and that were created because the Allow thread allocation beyond maximum option is enabled are destroyed immediately and do not wait for the duration of the thread inactivity timeout. For EnterpriseOne deployments, the time spent creating and destroying threads is unlikely to be noticed.

### ***Transaction timeout***

The online help describes this setting as “the number of seconds to allow a transaction to proceed before aborting it because it is taking too much time.” It is referring to a WebSphere orchestrated database transaction.

The EnterpriseOne servlets manage their own database transactions and there are relatively few of those. The vast majority of database transactions originate from the EnterpriseOne server. See the discussion in 4.1, “pSeries benchmark” on page 48, regarding the database transactions that originate from WebSphere and those that come from the EnterpriseOne server. This parameter is unused in EnterpriseOne deployments.

### ***Transaction inactivity timeout***

The online help describes this parameter as “the number of milliseconds a transaction can remain inactive before it is aborted.” It refers to a WebSphere orchestrated database transaction. Like the Transaction Timeout parameter, this parameter is unused in EnterpriseOne deployments.

## **4.3 xSeries**

Another benchmark was performed for the xSeries by the IBM and Oracle International Competency Center (ICC). It was published in June 2003 as *IBM @server xSeries Performance and Tuning Tips for the J.D. Edwards Web Server* by Don Gaines of IBM.

This document discusses tests performed against an Intel xSeries 440-based front-end architecture with a pSeries 630 back end. A high-speed UNIX back end was selected to allow the front end to be stressed to its limits and Technology Foundation bottlenecks to be

identified. For appropriate xSeries tuning guidance and configuration values, you can find this document on the Web at:

<http://www.ibm.com/support/docview.wss?uid=tsslwp100361>

## 4.4 iSeries

As with the pSeries and xSeries, an iSeries benchmark was performed by the IBM and Oracle International Competency Center (ICC). It was published in January 2003 in the document *Web Client Tuning Tips for iSeries* by Gerrie Fisk of IBM. This document discusses tests performed on OS/400 V5R2 using WebSphere 4.0, IBM HTTP Server, ERP 8.0, and OneWorld X2, SP20, and SP21. For appropriate iSeries tuning guidance and configuration values, you can find this document on the Web at:

<http://www.ibm.com/support/docview.wss?uid=tsslwp100284>

## 4.5 Miscellaneous tuning tips

Several miscellaneous tuning tips were uncovered during the course of stressing the system. These tips are provided in the following sections.

### 4.5.1 Database configuration changes

The majority of essential configuration changes on the pSeries benchmark project are database related. In fact, all advancements from 6,000 users on up to 15,000 users pertain to database configuration changes. You must be familiar with these concepts:

- ▶ Package locks
- ▶ Bufferpool configuration
- ▶ Running statistics

Use care if running statistics against empty tables that will be filled during the course of a test.

- ▶ Locksize changes for read only tables
- ▶ Append mode for tables that are primarily insertion tables

These are listed in order of the magnitude of improvement to the benchmark, measured by increases to the total number of users the system was able to service.

### 4.5.2 Watching for contention on the next numbers table

Several operations in EnterpriseOne require a unique number that is never reused. The *next numbers tables* are used to support this capability. The table is locked during the transaction to avoid duplicate numbers. In a high volume environment, this table can become a bottleneck.

EnterpriseOne provides an ability to divide this single table into multiple tables and assign groups of users to each table to avoid contention. Consult your EnterpriseOne documentation for details.

### 4.5.3 Ensuring your test scripts do not lock tables

In EnterpriseOne, end users can perform certain tasks that cause a “select for update” SQL statement to be sent to the database. This command causes a table to be locked. If the automated test tool script enters “think time”, the table is locked until the think time ends.

### 4.5.4 Paying attention to how your test tool classifies dropped transactions

When a transaction runs into problems, the test tool often classifies the transaction into categories that can tell you from where the problem originated. For example, when LoadRunner encounters a dropped transaction, it stops the virtual user and classifies the error into one of two categories: *Failed* or *Stopped*. Experience has shown that a Stopped transaction is usually a problem with the LoadRunner script, the data LoadRunner used to assemble the request, or the data in the database. Failed transactions are usually solved with configuration changes to WebSphere, EnterpriseOne, or the database.

Archived



## Managing Technology Foundation

This chapter outlines the best practices for maintaining and managing Technology Foundation.

## 5.1 Installation and maintenance roles

Table 5-1 lists some of the skill sets required to install and maintain the proposed architecture. While a single person can certainly perform more than one role, the skill sets are distinct.

Table 5-1 Skill sets to maintain Technology Foundation

Role	Skill	Reason
WebSphere administrator	Basic WebSphere Application Server administration	Administrator must be able to establish clones and server groups to understand and alter Extensible Markup Language (XML) configuration files, institute runtime monitoring, understand resource analyzer, perform problem isolation for application servers, deploy EAR files, and install patches.
	WebSphere Application Server workload and cluster management expertise	Administrator must be able to configure for high availability.
	WebSphere Application Server Network Deployment Management Expertise	Needed for WebSphere Application Server Version 5 or later (meaning Technology Foundation Version 5 or later).
	WebSphere Application Server tuning expertise	Administrator must know how to configure threads, connection backlogs, servlet caching, Dynacache, etc.
	Basic HTTP Administration	Administrator must understand how Hypertext Transfer Protocol (HTTP) plug-ins work to link WebSphere Application Server with the HTTP server.
	HTTP Tuning expertise	Administrator must understand how to configure Apache daemons and threads.
Database administrator	DB2 tuning expertise	For Technology Foundation (particularly in Version 4), there is additional WebSphere Application Server database configuration and tuning, beyond normal EnterpriseOne back-end server database administration.
Network and security administrator	Basic network administration	Configuration of Technology Foundation machines, Domain Name System (DNS), IP addressing, etc. is necessary.
	Firewalls, threat detection, reverse proxy, threat analysis	Administrator must understand how to configure the firewalls and reverse proxies. Any mistakes in this domain can expose the system to security threats. The administrator must know how to effectively monitor the network and perform active threat analysis and response.
	Authentication mechanisms and authorization models	Administrator must understand how LDAP is used. The administrator must know how to establish secure links between WebSphere and Lightweight Directory Access Protocols (LDAPs) such as Active Directory, IBM Directory Server, or Domino.
Portal developer	Portal Development	If Portlet customizations are desired, portal development expertise is needed.
Portal administrator	Portal Administration and Access Control	Administrator must understand basic Web modules and WAR files, deployment techniques, portal security (user and group management), scripting, performance monitoring
	Portal Web Clipping, Themes and Skins, Cascading Style Sheets	If plans exist to customize the look and feel of Web pages to conform to the corporate standard look and feel.
	Portal Failover	Administrator has an understanding of Portal failover particularly as it pertains to Domino, Lotus Instant Messaging and Web Conferencing, and Lotus Team Workplace and using clustering in conjunction with Network dispatcher to achieve failover.



Role	Skill	Reason
Lotus Collaboration administrator	Lotus Instant Messaging and Web Conferencing	Administrator must be able to establish instant messaging and person awareness.
	Lotus Team Workplace	Administrator must be able to establish a collaborative work area.
	Lotus Domino	Administrator has an understanding of LDAP authentication and performance monitoring techniques. This is especially useful if you are using Domino as the LDAP server.
System administrator	Administrator who has an understanding of the system as a whole	Administrator must be able to trace problems across application boundaries and physical machines, recommend physical topology changes to address performance problems, and institute cross-tier performance monitoring mechanisms.

## 5.2 Operations monitoring and management tools

Each product in the Technology Foundation suite offers several monitoring and management tools. Each section that follows discusses the tools that are offered for the given product.

### 5.2.1 WebSEAL

Reverse proxy mechanisms, such as WebSEAL, are devoted to preventing security compromises as well as to detecting and reporting on them. WebSEAL provides the ability to alert administrators to problems. It automatically prioritizes and responds to events accordingly with actions ranging from simple e-mail or pager notification to shutting down entire portions of the network.

WebSEAL provides a highly configurable event manager that allows the viewing and management of potential security problems. It has the ability to consolidate security information from a several different sources (even from Norton Antivirus software running on corporate PCs) and filter the real alerts from innocuous events. Since security information is continually logged, reporting packages can be used to generate canned or custom reports.

For information about the monitoring and reporting capabilities of WebSEAL and its supporting security software, see *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014.

### 5.2.2 Network Dispatcher

Monitoring the Network Dispatcher component can be done through various mechanisms. The *ndcontrol* utility can produce useful ASCII text reports, although they are rough looking. The utility uses several command line options, such as manager report, high status, or high takeover, to determine the overall system status or the current state of failover. These reports are essential to run when Network Dispatcher is installed. They help to ensure that requests to the HTTP servers are properly distributed and to the various HTTP server in the proportions that were configured. You must view the reports when failover testing is performed.

**Note:** Request distributions may be evenly weighted across HTTP servers, but not necessarily so. One of the HTTP servers can be a vastly superior machine that is capable of handling far more requests.

Network Dispatcher also provides a Server Monitor capability that graphically displays the load across all machines in an elegant three-dimensional bar chart. The chart can be configured to update the information over larger or smaller time samples. It can also be configured to show the load based on various metrics such as how busy the HTTP server operating system is, how many new connections are submitted to each server, how loaded each HTTP server says it is, and many other metrics. The Server Monitor chart provides a consolidated view of all HTTP servers.

Given the back-end bottlenecked nature of EnterpriseOne traffic, the Network Dispatcher and the HTTP servers are likely to easily handle all requests if they are at all reasonably sized. For Technology Foundation, Network Dispatcher monitoring is primarily needed to ensure proper installation, to simply check that any dispatcher component is not down or in failover, and to give administrators a consolidated metric for the number of requests getting submitted across all servers.

You can learn more about Network Dispatcher monitoring in the redbook *WebSphere Edge Server: Working with Web Traffic Express and Network Dispatcher*, SG24-6172. In the same redbook, refer to the “High Availability” section and discussions about the proper techniques to configure failover for Network Dispatcher.

### 5.2.3 HTTP

The IBM HTTP Server (powered by Apache) is built upon the Apache server—the most widely used HTTP Server on the Internet. Therefore, there are numerous monitoring tools for the application provided by Apache and third-party vendors. Administrators are confronted with products that can display fancy graphs that address such questions as:

- ▶ How many pages were returned with successful status codes? What percentage were 404s (page not found), 200s (successful), or 500s (internal server errors)?
- ▶ What is the response rate per second, shown over time?
- ▶ What is the throughput? How many pages per second are downloaded?
- ▶ How many retries are getting submitted?

While all of these metrics can help to verify a problem exists, for Technology Foundation, the HTTP server is seldom the source of problems. HTTP servers are designed and instrumented to support large numbers of transactions per second. However, EnterpriseOne Technology Foundation users typically do not deluge the server with many short requests. Instead, they tend to reach the servers with a few long running back-end bottlenecked requests. Many of the performance monitoring capabilities for the HTTP server are oriented around a somewhat different usage model than that of Technology Foundation customers. Ironically, the abundance and variety of diagnostic information can inadvertently hide problems in a deluge of information. It is important to know where to begin to look for likely trouble sources.

Much of the monitoring information for the HTTP server pertains to speed. For Technology Foundation, this is unlikely to be an HTTP server problem since the front end can process requests much faster than the back end, which must transact to disk. Nevertheless, it is important to monitor the end-to-end response times that users receive. A sudden and dramatic spike in average response times is likely to trace to back-end server problems.

Many of the standard performance metrics, for Technology Foundation, are relatively uninteresting on a day-to-day basis. However, they can be helpful after a new installation or an upgrade. If users encounter numerous “page not found” errors after an upgrade, a hyperlink may be misdirected or an Hypertext Markup Language (HTML) page may be missing. After an upgrade, response rates can be checked against previous values to ensure that the system is behaving as expected.

For day-to-day monitoring (not after a new installation or upgrade), administrators must focus on simple error log files, which can be automated. See 5.2.8, “System monitoring” on page 76.

When a problem occurs in WebSphere or in the Collaborative Portal, the errors are reported to the user in terse “Internal Server Error” Web pages that each have a status code in the 500 to 599 range. Error codes in this range generally indicates a problem on the WebSphere Application Server. This is not to say that a 500 error means a WebSphere server has gone down. It can simply mean the WebSphere server has become overloaded, its threads are exhausted, its operating system is unable to buffer any more HTTP packets, or a host of other sporadic errors. Errors in the 500 to 599 range call for investigation and should not be ignored.

**Note:** These problems are not to be confused with a problem in the back-end server that is packaged into a Web page with a user friendly message stating the nature of the problem.

Technology Foundation customers using any highly secure logical architecture should be aware that WebSEAL reverse proxy mechanisms hide the true originators of requests. Consequently, the proxy masks the reporting capabilities designed to show information about the sources behind HTTP requests. Therefore, the information must be provided elsewhere.

## 5.2.4 WebSphere Application Server

WebSphere provides many monitoring mechanisms that can yield a wealth of information. You can learn more about each in *IBM WebSphere V5.1 Performance, Scalability, and High Availability: WebSphere Handbook Series*, SG24-6198. However, in a Technology Foundation environment, much of this information is superfluous, particularly if the system is tuned properly.

For example, if the min and max thread counts are set equal to each other (as recommended in Chapter 4, “Configuring and tuning Technology Foundation for high transaction volumes” on page 47), then the vast amount of information that WebSphere provides on dynamic thread creation is unnecessary and administration is greatly simplified. Similarly, the information returned regarding Enterprise JavaBeans (EJBs) is not yet used. The database transaction timeouts do not apply at this level because the EnterpriseOne Web Server application manages its own timeouts. Consequently, much of the information reported can confuse administrators and lead them to misinterpret the source of problems.

For Technology Foundation, a good, straightforward place to check for problems is in the log files. Examining log files frequently helps to find problems that otherwise go unnoticed. See 5.2.6, “EnterpriseOne Web Server (JAS)” on page 74.

You must familiarize yourself with the following sets of files:

- ▶ /usr/WebSphere/AppServer/logs/\*

The native.log file in this directory deceptively belongs to the HTTP server and not to WebSphere. See Chapter 4, “Configuring and tuning Technology Foundation for high transaction volumes” on page 47, for tips to configure MaxClients and Allow thread allocation beyond maximum, as well as the potentially disastrous but not immediately noticeable problems that arise when deleting all of the log files from this directory.

- ▶ /usr/HTTPServer/logs

The access.log records all transactions to the HTTP server and can be optionally turned on in the httpd.conf configuration file.

- /usr/EnterpriseOne/JAS/EnterpriseOne.ear/webclient.war/logs

The JAS servlet output resides here.

While the documents named in “Related publications” on page 113 discuss the precise configuration parameters for the generation of these files, it is important to realize that all of the log generating processes can be set to report more or less information. For example, the server can be configured to log informational and debug messages (certainly not the default).

The Tivoli Performance Viewer (or Resource Analyzer as it was previously named) can provide some useful information although most problems are visible in the log files. Thread levels can be checked under peak load to ensure thread limit violations are not imminent.

**Note:** Thread limit violations cause the Java Virtual Machine (JVM) to be marked as busy and the HTTP server logs a message in the native.log file stating that the server is attempting to redirect the request to another application server.

The Performance Viewer can also monitor the frequency of garbage collection. However, garbage collection, typically an all in-memory operation, is unlikely to incur a noticeable duration with Technology Foundation. JVM memory consumption is worth monitoring as well.

WebSphere’s Performance Viewer is primarily useful, in an EnterpriseOne deployment, to monitor the frequency of garbage collection, to monitor thread pool usage, and to check JVM memory consumption. In this sense, it is a tuning tool and is generally not useful for noticing or diagnosing dropped transactions. If transactions are dropped, they are noticed by the HTTP server. See Chapter 4, “Configuring and tuning Technology Foundation for high transaction volumes” on page 47, for tips about MaxClients and Connection backlog.

Many third-party monitoring tools exist, but the monitoring needs for the WebSphere portion of Technology Foundation are relatively modest. This layer of the application is simple. It is better for users to avoid the overhead of such tools unless they can identify a specific need that is not met by the Tivoli Performance Viewer. Many of the third-party monitoring tools are more appropriate for development than monitoring since they tend to be so invasive. A tool that reports what methods are executed is extremely valuable to developers. However, it is questionable whether most client administrators find method information useful or are willing to suffer the performance degradation necessary to leave such a tool turned on all the time.

### 5.2.5 Portal

For Technology Foundation, managing the Portal has more to do with configuring Secure Sockets Layer (SSL), maintaining the credential vault, and managing customization information than it does with performance and operation monitoring. Portal provides administration portlets to examine information about users who are currently logged in. If required, tracing information can be turned on to debug portlet problems.

### 5.2.6 EnterpriseOne Web Server (JAS)

EnterpriseOne provides a tool to monitor information in the servlets. Server Administration Workbench reports much of the same information that is available from WebSphere’s monitoring tools, but with a few important additions. Since the EnterpriseOne Web Server application manages its own database connection pools, Server Administration Workbench is the only place to monitor the usage of these pools. The EnterpriseOne Web Server application also manages virtual threads and virtual thread groups. It maintains the mapping from a large number of Web clients to a limited set of back-end kernel processes. Oracle

support staff may ask administrators to report information from the Server Administration Workbench displays.

You can find extensive documentation about Server Administration Workbench in your EnterpriseOne user manuals. The Oracle PeopleSoft Customer/Partner Connection also contains helpful information about Server Administration Workbench, which you can find on the Web at:

<http://www.oracle.com/peoplesoft/integration.html>

## 5.2.7 DB2 Universal Database

Since operations performed in memory take nanoseconds, while disk operations take six orders of magnitude longer, database optimization is typically more beneficial than any other performance tuning area. This is by far the most important area for performance monitoring.

However, for Technology Foundation, performance gains from database optimization depend greatly on which database instance is optimized. A database that is rarely accessed or that can store virtually all information in an in-memory buffer pool shows little benefit from optimization. This is certainly the case of the WebSphere 4.x repository. WebSphere 5.x uses XML files rather than the database to store repository information. Since EnterpriseOne does not use Web-based session failover mechanisms, the WebSphere database is rarely accessed by the application server.

Similarly, the LDAP database does not require much performance or operations monitoring, other than simply ensuring that the database is up. The LDAP database serves small chunks of text-based information that is primarily read only. Frequently the entire database can fit in memory, and with the absence of updates, the cache hit rate is very high. Moreover, the LDAP database comes pre-optimized for read operations. The Portal server incurs some minor database traffic as well. Personalization information and user preferences are stored in the portal database.

The back-end EnterpriseOne database is a very profitable target for performance monitoring and optimization. One of the allowed Technology Foundation architecture customizations is to consolidate the Technology Foundation databases on the back-end server. This customization has the advantage of minimizing maintenance since another database instance running on a separate machine is not called for. However, it has the downside of blurring the distinction between back-end performance and front-end Web database issues. If the back end is significantly accessed, front-end performance can suffer. In such scenarios, administrators can easily mistake the true source of problems.

Database tuning is the subject of many books and consequently beyond the scope of this redbook. Snapshots, event monitoring, and log analysis are useful tools that all database administrators should understand. It is important to know how indexes can profoundly increase (and actually decrease) performance. Administrators should know how to see which indexes are used (not necessarily straightforward) and explain the plan for frequent Structured Query Language (SQL) statements. They should know how to reset and properly generate the statistics that govern index selection.

**Note:** A common mistake is to generate statistics on empty tables such as the working tables that EnterpriseOne periodically deletes. The database needs actual data in the table to determine an appropriate indexing scheme. Generating statistics on an empty table can cause the exact opposite of the desired results by inadvertently forcing the use of the wrong index.

It is true that databases are worthy of monitoring a prime source of performance problems, but only if they are actually reached. Many of the Technology Foundation databases have very few transactions. Nevertheless, we recommend that you configure and run the Health Monitor on all DB2 instances.

## 5.2.8 System monitoring

Many sophisticated IBM Tivoli brand tools exist to monitor the health and safety of the system as a whole.

**Note:** Because you can use Tivoli tools to monitor far more than just Technology Foundation, and because they are of tremendous value to EnterpriseOne and all corporate enterprise applications, they are not included as part of the Technology Foundation offering. You must purchase them separately.

Tivoli Enterprise™ Console provides a composite view of how all the nodes in the overall system are operating. The Enterprise Console allows administrators to view complex, interconnected systems as a whole, even though they are composed of disparate middleware, database, and applications. Enterprise Console has adapters for each of the Technology Foundation platforms—OS/400, AIX (UNIX), and Windows—as well as numerous other types of applications, operating systems, and hardware devices. Enterprise Console diagnoses root problems and allows reaction before the problems become worse. In many cases, the resolution to problems is automated and reactions are processed according to a configurable rules tree.

Event adapters gather information and funnel it up to the IBM Tivoli Enterprise Console®. These collectors include monitoring for business integration, Web tier components, databases, network monitoring, storage area network (SAN) and redundant array of independent disks (RAID) systems monitoring, firewall and intrusion detection, etc. Tivoli Enterprise Console is the recommended Technology Foundation monitoring solution because it is pre-integrated with the products that compose the logical architectures. For more information, refer to:

<http://www.ibm.com/software/tivoli/products/enterprise-console>

## 5.3 Maintenance

The following sections discuss the best practices for maintaining and upgrading Technology Foundation. While end users are not in the business of writing code for Technology Foundation, this does not mean that many of the principles of software development do not apply to their environment, especially since EnterpriseOne is highly customized and tailored to the exact business needs of end users. Taking an upgrade requires a quality assurance environment, a staging environment, and possibly a performance testing environment. Establishing this environment is the topic of the first subsection. The second section deals with commonly overlooked maintenance practices.

### 5.3.1 Quality assurance environments for new software releases

A high quality of service architecture means being careful about how you change and update software. Supporting a production environment requires a robust quality assurance environment. The purpose of this section is to propose a supporting architecture infrastructure that is appropriate to upgrade and maintain Technology Foundation.

A common mistake is to assume that only enough hardware to construct the end production environment is needed. Minimally, you must establish a Technology Foundation quality assurance environment to test upgrades. Moreover, clients who write and install their own EnterpriseOne customizations should conform to the standards of software development even if they are not writing code that “compiles” in the traditional sense.

If an upgrade has an unexpected and serious bug, the system cannot be restored to its previous state if that previous state does not exist. A history of changes cannot be determined if a configuration management system does not exist. Development, customization, and testing simply cannot take place on the live production environment. Consequently, an enterprise quality infrastructure includes configuration management, defect tracking, and isolated environments that protect users from testers and testers from EnterpriseOne development and customization.

The following sections present an overview of an infrastructure architecture capable of properly supporting Technology Foundation.

### Symbol conventions for figures

For simplicity, in this section we use a simple, double layered symbol to represent the entire Technology Foundation highly available architecture (in either the standard or highly secure configurations). See Figure 5-1.

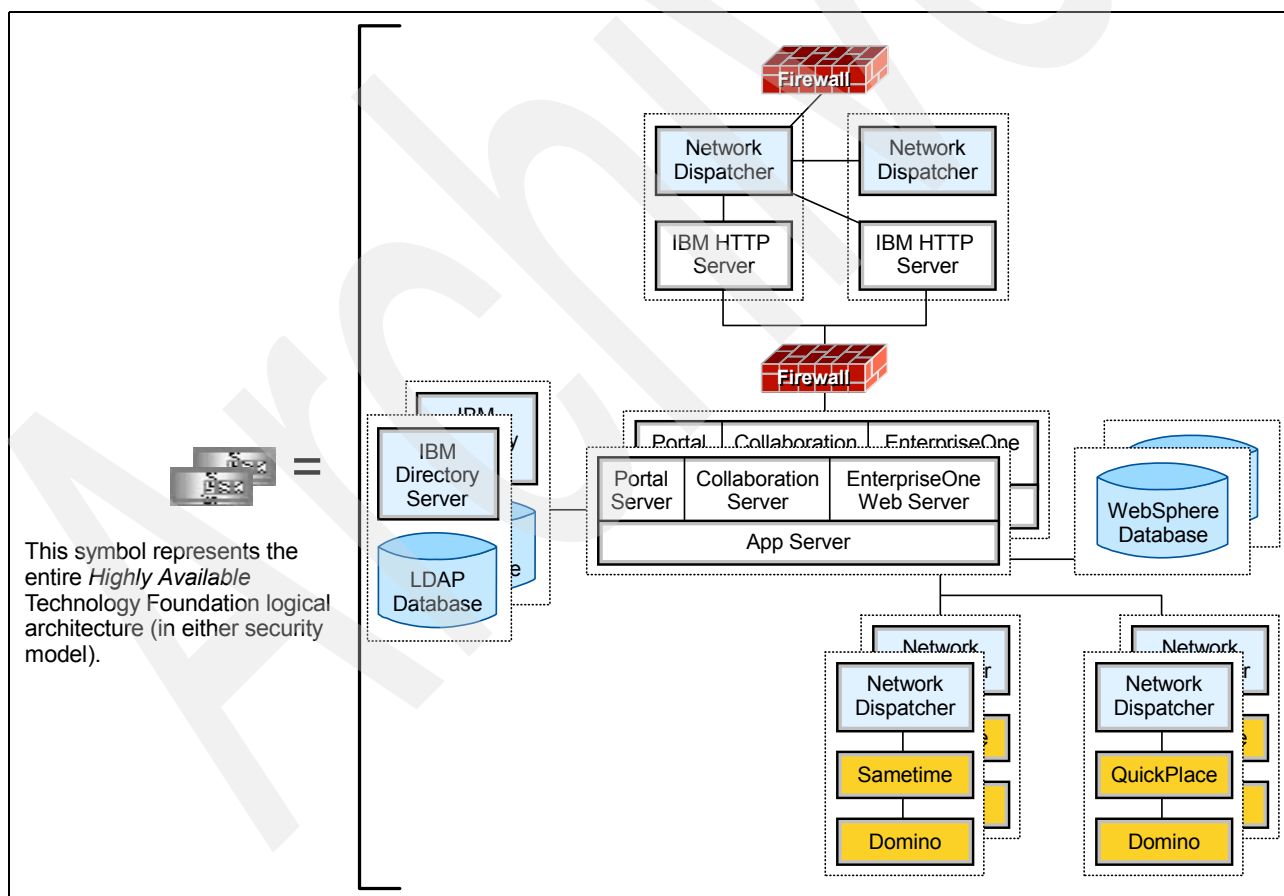


Figure 5-1 Technology Foundation double-layered symbols

Similarly, we use a single, triple layered symbol to represent the entire continuously available Technology Foundation architecture (in either the standard or highly secure configurations). See Figure 5-2.

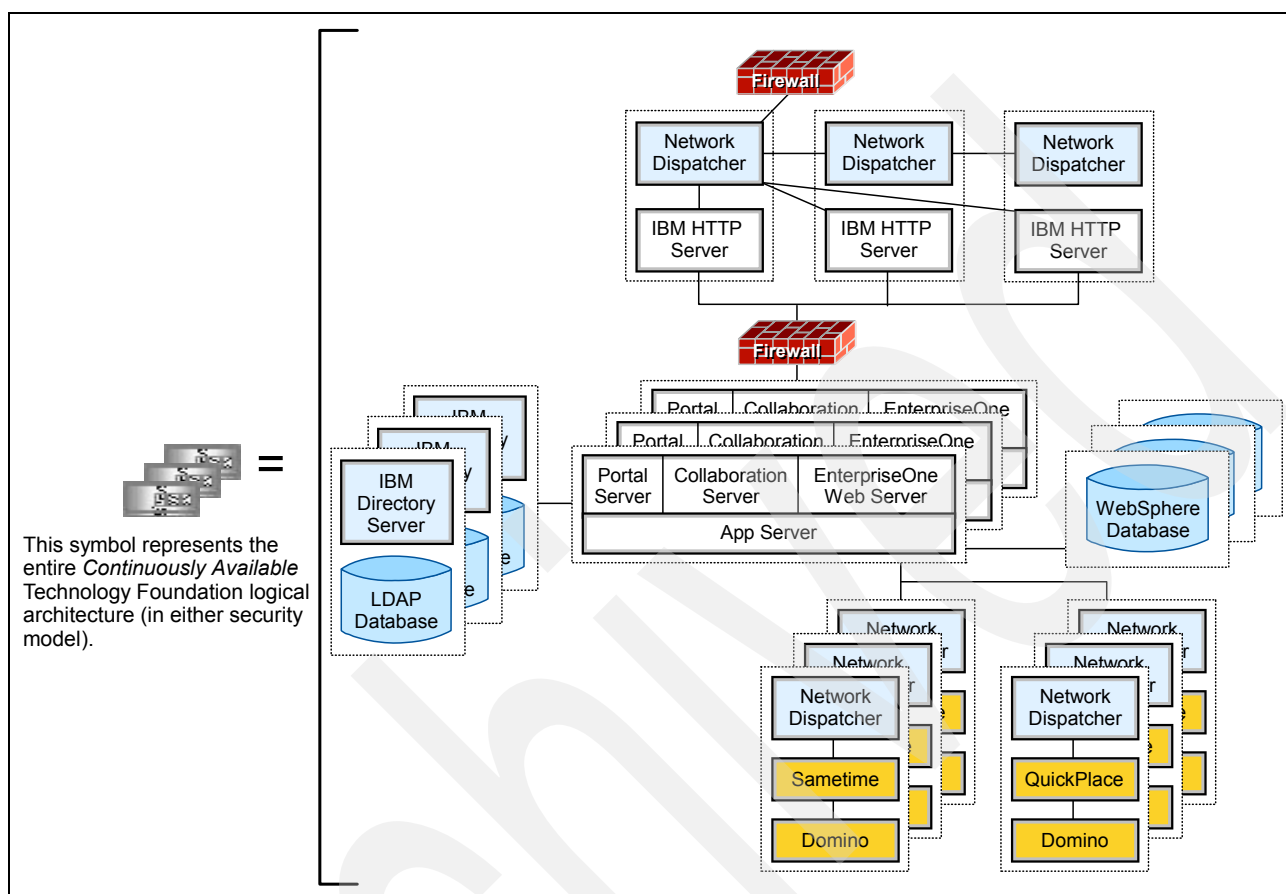


Figure 5-2 Technology Foundation triple-layered symbols

It is important to remember that the simple symbols from these two figures are used extensively in the following sections and expand to those entire Technology Foundation configurations.

Different types of hardware may be required to support infrastructure. For example, even in a UNIX environment, many clients prefer to host their configuration management software on PCs. Development also frequently occurs on PCs. Because these requirements are so common, the example in the following section assumes a two-platform infrastructure. Figure 5-3 illustrates the symbol conventions used to support that example.

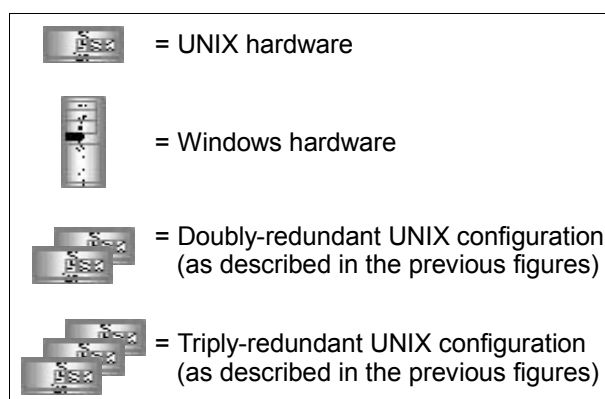


Figure 5-3 Technology Foundation symbol conventions



## Sample infrastructure topology

Figure 5-4 shows a high level view of a Technology Foundation infrastructure environment that is capable of supporting configuration management, defect tracking, performance testing, normal quality assurance, and parallel development. Some clients do not require all of the components in this architecture depending on their degree of development effort.

For example, if no custom portlets are written and only out-of-box EnterpriseOne modules are used, then the entire development layer can be forgone. The Integration Layer can be eliminated if only a single development team exists and has an integration server that is made of the same hardware as the final production environment. However, you must first understand the layers before you discard large sections of the infrastructure.

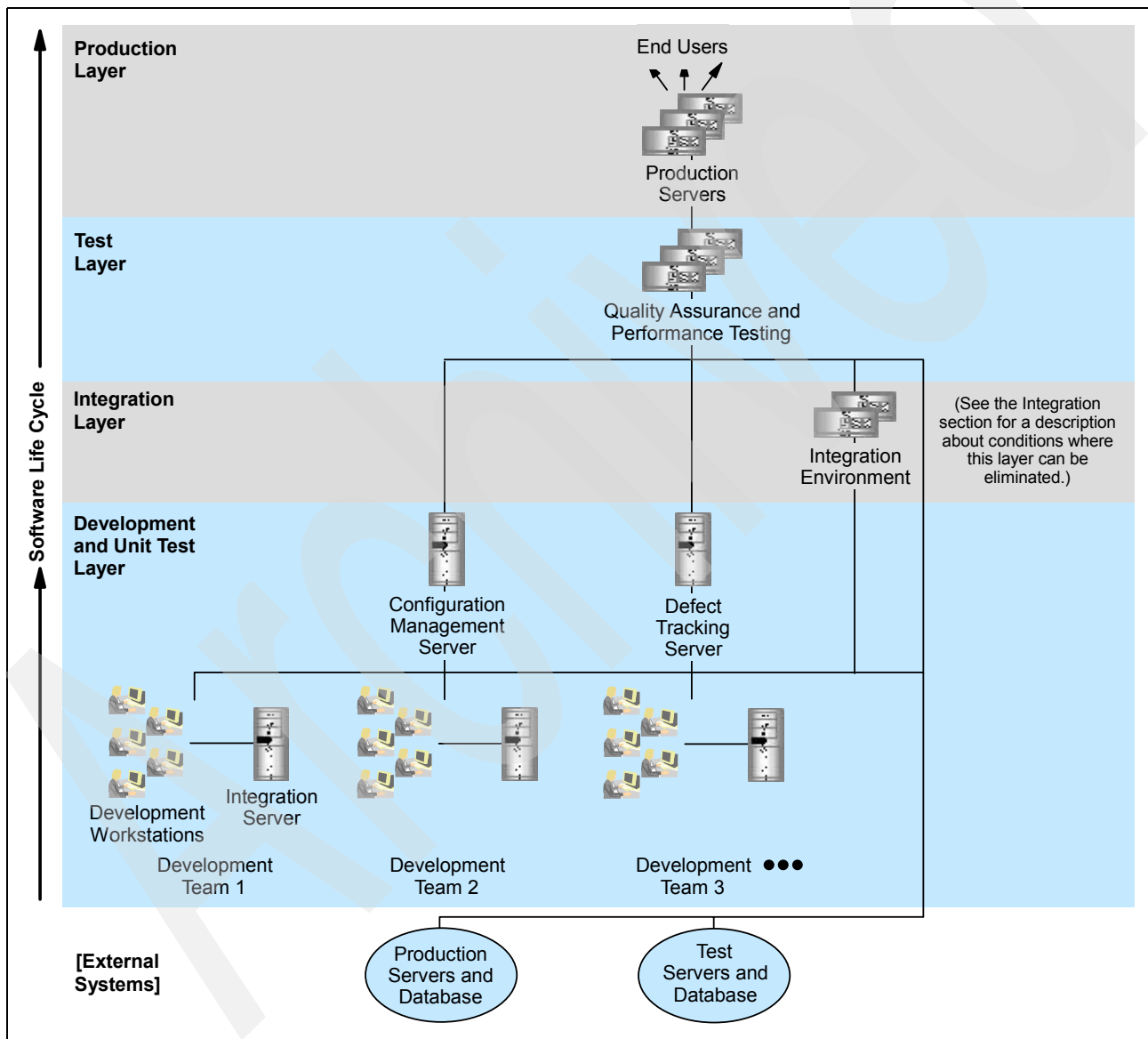


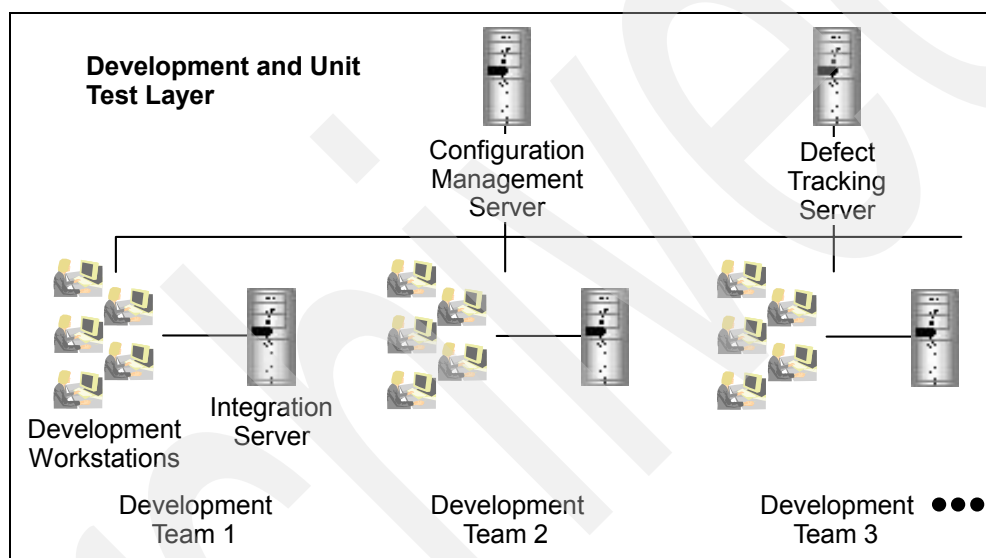
Figure 5-4 Technology Foundation infrastructure

The individual details of each infrastructure layer are discussed in detail in the following sections. The overall infrastructure directly maps into the life cycles of software development. For example, the bottom layer, Development and Unit Test, represents the environment that supports coding and unit testing. The output from the Development and Unit Test Layer is

consolidated in the Integration Layer. Such software as custom portlets or EnterpriseOne customization is tested here, by lead developers, prior to delivery to the testing staff. In a complex development environment, the Test Layer can be divided into multiple channels that correspond to emergency fixes, small projects, large projects, or performance testing. However, for this example the Test Layer remains simple and has only a single environment that is used for all of these purposes. The software is ultimately moved from the Test Layer into actual production as it begins to service the enterprise.

### ***Development and Unit Test Layer***

The Development and Unit Test Layer (Figure 5-5), the bottom layer, represents the environment that supports the coding and unit testing that occurs with composing portlets or customizing EnterpriseOne modules. While clients do not code in the traditional sense of compiling and producing executables, the configuration and customization model of EnterpriseOne requires a professional development approach, including tracking changes in a configuration management system and managing the history of bugs.



*Figure 5-5 Development and Unit Test Layer*

Development of new portlets or new applications made in the Technology Foundation environment similarly requires an organized and well managed development approach. See 5.5, "Beyond Technology Foundation" on page 84.

This layer is divided into project teams that may be organized by engineering expertise, for example, teams that specialize in HTML and JavaServer Pages (JSPs), or that may be organized by product, such as Customer Management or Scheduling. In small organizations or organizations with modest development requirements, there may be a single team or no team at all.

If there are no development teams, or if there is only a single team that integrates their work on an integration server that has the same hardware configuration as the production and test environments, then the entire integration layer is not needed.

The integration servers at this level are not redundant. Redundancy is typically not tested until software reaches higher layers. Similarly, the configuration management and defect tracking servers do not need CPU redundancy but do need data redundancy. These two servers require high availability disks, and corresponding backups and off-site storage, since the loss of code can be catastrophic.

Since the software configuration of the machines in this layer differs substantially from the standard configuration found in all other layers, you should have a general idea of the software that may reside on typical machines in this layer. Naturally, the configuration of these machines differs if a client performs EnterpriseOne customizations as compared to a different client who augments the WebSphere Application server with custom Java 2 Platform, Enterprise Edition (J2EE) applications or portlets. For example, suppose a client augments the WebSphere Application Server with a custom application. See 5.5, “Beyond Technology Foundation” on page 84, to learn about the advantages of doing this.

For such a client, Table 5-2 provides a possible software configuration for each machine in the Development and Unit Test Layer.

Table 5-2 Configuration requirements

Component	Software
Developer's PC	<ul style="list-style-type: none"> <li>▶ WebSphere Studio Application Developer, Version 5.x</li> <li>▶ ClearCase® LT (comes with WebSphere)</li> <li>▶ ClearQuest® Client</li> <li>▶ Windows 2000 Professional with Service Pack 2 or later, or Windows XP Professional, or Windows NT® Workstation or Server V4.0 with Service Pack 6a or later</li> <li>▶ Windows - Microsoft Internet Explorer® 5.5 with Service Pack 1 or later, or Netscape Navigator V4.76 or later</li> </ul>
Integration Server	<ul style="list-style-type: none"> <li>▶ WebSphere Application Server, Version 5.x</li> <li>▶ DB2 7.2, Fix Pack 7</li> <li>▶ IBM HTTP Server</li> <li>▶ Windows 2000 with Service Pack 3 or later</li> </ul>
Configuration Management Server	<ul style="list-style-type: none"> <li>▶ ClearCase Server (comes with WebSphere) or comparable product</li> <li>▶ Windows 2000 with Service Pack 3 or later</li> </ul>
Defect Tracking Server	<ul style="list-style-type: none"> <li>▶ ClearQuest Server</li> <li>▶ Windows 2000 with Service Pack 3 or later</li> </ul>

You can learn about the configuration requirements on the Web at:

<http://www.ibm.com/software/ad/studioappdev/sysreq>

### Integration Layer

The Integration Layer (Figure 5-6) is simple since it is intended for short-lived integration testing. For example, a client who intends to enhance Technology Foundation with custom enterprise applications coordinates the work products from their separate teams on these machines. See 5.5, “Beyond Technology Foundation” on page 84.

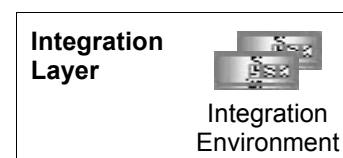


Figure 5-6 Integration Layer

A different client performing a high degree of EnterpriseOne customization by multiple teams consolidates the work products also in this Integration Layer. All software deployed to this layer is tracked in the configuration management server in the Development and Unit Test Layer. The Integration Layer provides the last stop for developers before the software is handed to testers in the Test Layer.

**Note:** Strictly speaking, emergency fixes may pass directly from the Development and Unit Test Layer directly to the Test Layer for regression testing. This direct line between the development and test layers is justified since significant integration work is not part of a simple emergency fix.

In rare cases, clients who develop for multiple platforms have two or more sets of different hardware in this layer, although most clients target a single production environment.

The underlying hardware platform composing the Integration Layer may differ from the development layer since the Integration Layer must closely match the final production environment. Development often takes place on PCs even if the final platform is, for example, an iSeries server. This is typical of J2EE development environments and is one of the benefits of the technology. However, developers must test their software in an environment that is similar to the final production environment before handing over the software to be tested. The Integration Layer serves this purpose.

The integration environment is minimally a doubly redundant configuration. That is, the integration environment completely simulates a full production environment, although not necessarily the final production environment, which may be triply redundant. Redundancy in the Integration Layer is highly desirable, and not because high availability is required. It simply simulates the actual production environment and begins to drive out issues associated with redundancy.

### Test Layer

For the purposes of this simple infrastructure architecture, the Test Layer is kept inexpensive and simple by consolidating several disparate purposes into a single environment (Figure 5-7). In this environment, all testing takes place on a one single set of machines. This environment is used to perform several functions that, in a more complex development shop, may otherwise be divided into separate environments. These functions include normal quality assurance testing for small and large projects, performance and stress testing, and pushing out emergency fixes.



Figure 5-7 Test Layer simple development environment

It is important for the test environment to perfectly match the production environment—even with the same level of redundancy. This is because the test environment must be able to accurately model performance and behavior under load. Bugs and performance problems pertaining to redundancy are not observed if the environment is not redundant and in the same configuration as the production environment. The Test Layer may or may not have its own configuration management machine since every test department has its own preferences for controlling source code.

While this simple configuration requires minimal hardware and keeps quality assurance costs low, it has some drawbacks. For example, if an emergency fix must be pushed out in the midst of normal testing for the next large release, one version must be stored and unloaded while the software for the emergency fix is pushed through. By contrast, in a busier, more complex development environment, the Test Layer might appear as illustrated in Figure 5-8.

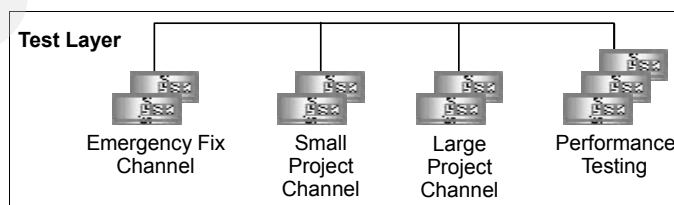


Figure 5-8 Test Layer complex development environment

Here the test layer is divided into four *channels*. The idea of each channel allows for simultaneous testing against unique baselines. The *emergency fix channel* is designed to quickly cycle development fixes into the production environment. The *small project channel* is designed for small-to-medium sized releases, which require moderate testing. The *large project channel* is for major releases, which require the entire suite of all testing procedures and regression tests. A separate performance testing area is also provided.

Strictly speaking, the performance testing environment is not considered a channel since testers and developers move code here optionally and temporarily to test performance and that software may not be pushed into production. Each channel is redundant because it must simulate an actual production environment and drive out any bugs associated with failover.

**Note:** A notches pattern of usage of the Emergency Fix Channel makes redundancy optional.

Most Technology Foundation end users establish a Test Layer that is simple. It is possible to compromise between the two extremes presented in this section by selecting only one or two delivery channels.

### **Production Layer**

The architecture and configuration of the Production Layer (Figure 5-9) is the subject of the first part of this redbook. It is the layer that serves the actual end users. It must be secured both from the internal corporate network and from the test and development environments with the use of reverse proxies and firewalls.

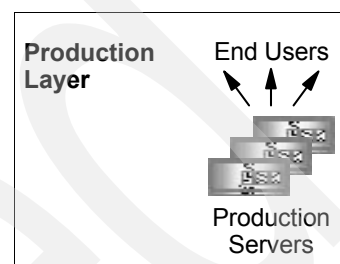


Figure 5-9 Production Layer

## **5.4 Commonly overlooked best practices for maintenance**

Technology Foundation maintenance best practices are about providing a hardened, enterprise quality infrastructure. A comprehensive treatment of maintenance best practices for enterprise systems in general is beyond the scope of this text. However, some commonly overlooked practices are worth mentioning, especially in the context of Technology Foundation.

### **Anticipate increased demand**

Actively monitor resource consumption and the general performance of the system under peak and sustained loads. Be proactive. Anticipate demand for additional capacity. Do not wait for disaster to strike. Establish well defined thresholds for all system resources and identify all criteria for determining when that threshold has been reached. Don't engineer solutions at the last moment in response to an emergency. Have documented, predefined mechanisms to add capacity. Know how and when vertically and horizontally scaling strategies will be used. Database growth, CPU consumption, Web tier traffic, network activity, etc. all must be intelligently monitored for potential problems.

### **Take backups**

Enterprise applications are mission critical. Off-site storage of backups is essential. Backup strategies should include disk images, incremental backups, full backups, simple file copies, and other mechanisms as appropriate. The depth and frequency of the backup are determined by comparing how frequently the information changes, how long it takes to back up, and how much system operations can be disrupted by the backup procedure itself against the cost of lost information.

## **Security components need maintenance too**

Maintenance best practices pertaining to security include:

- ▶ Make sure operating systems are kept up to date with the latest patches, fixes, and service packs. This is particularly important for PCs.
- ▶ Close all unused accounts.
- ▶ Turn on logging and audit all transactions as resources permit.
- ▶ Turn off unnecessary operating system services and disable unused applications.
- ▶ Provide automated intrusion detection.
- ▶ Ensure physical access to the enterprise machines and network is limited, monitored, and logged.

## **Don't take power for granted**

Uninterruptible power supplies are required for mission-critical systems. The quality of the power supply is determined by weighing the client's unique requirements against the cost of downtime. Know the general reliability of power in the area. Understand the true cost of system downtime (see "The business case for high availability" on page 39). If it is critical that the business remain up even during a natural disaster, ensure that the uninterruptible power supply capacity is capable of sustaining the environment for the duration of the most probably worst case down time. Power can be obtained from more than one substation and power generators can take over for uninterruptible power supply batteries during extended outages.

Monitor the system. It is essential to monitor the health and safety of the system, as described in the 5.2, "Operations monitoring and management tools" on page 71. Establish thresholds for all system resources. Also establish procedures for coping with resource consumption problems in advance, before disaster strikes.

## **5.5 Beyond Technology Foundation**

Now that you've established a high availability and high performance environment, what can you do to get the most value from it?

In addition to running your normal JD Edwards EnterpriseOne applications, it is possible to mix in custom portlets, MQ integration with external applications, or add more J2EE applications. You can integrate password authentication across multiple applications through Portal's Credential Vault capability. You can perform business process modeling with WebSphere's Enterprise Edition. You can optimize your application suite for remote access or to be publicly available through the Internet. This section discusses these potential directions.

### **5.5.1 Enhancing Technology Foundation with portlets**

You can easily enhance the Technology Foundation portal view with additional standard or custom-fit portlets for your organization. For example, organizations that have off-site daily activities may desire a weather portlet so that job managers can readily see the effect of local weather on their scheduled activities. Maps can display today's destinations. Portlets can announce the latest information from the CEO to all employees or they can display pertinent stock and financial information. Portlets can be used to support chat or provide team rooms for groups of employees. They can be used to edit and display information from applications such as Oracle Financials. IBM provides access to thousands of prepackaged portlets.

**Note:** If you have a restricted license agreement, check your JD Edwards Technology Foundation contract for limitations and restrictions on the types of portlets allowed. Some Technology Foundation restrictions are lifted with full Portal license purchases.

WebSphere Portal Extend comes with many useful standard collaboration portlets. These portlets include e-mail access, calendar information, to-do lists, team room services, document library access, and discussion boards. In addition to the standard components, Lotus Collaboration components provide platform-neutral, J2EE compliant code for clients with their own development communities. These Lotus Collaboration components make it easy for portlet developers to provide custom access to e-mail, instant messaging, and team rooms. Lotus Collaboration components allow developers to provide these features to end users without knowing the behind the scenes, server-specific details such as Lotus Team Workplace or LDAP server names.

Refer to the Collaboration chapter in *IBM WebSphere Portal V4.1 Handbook, Volume 1*, SG24-6883, for more information.

## 5.5.2 Portal's Credential Vault

Portal's Credential Vault capability allows end users to use a single password, requested once, to sign into multiple applications (including external, non-J2EE applications). Tracking numerous passwords to multiple back-end systems is an administrative headache. Ironically, multiple passwords frequently cause security risks, since end users tend to manage these difficult-to-remember passwords with non-secure techniques (such as writing them down somewhere in their office environment or storing them in a non-secure file). Credential Vault solves this problem by managing multiple back-end passwords for the user.

Credential Vault provides mechanisms where users can access multiple back-end systems with their unique logins, or where groups of users can share a single unseen back-end login to a corporate system. Hybrids of these two approaches are also supported, as well as private or shared passwords to Internet resources. Vaults can be user managed or administrator managed.

The "Credential Vault" section in the *IBM WebSphere Portal V4.1 Handbook, Volume 1*, SG24-6883, describes the extensive configuration options available.

## 5.5.3 Publicly accessible enterprises

Logical architectures capable of supporting large volumes of users and that provide public access to enterprise applications usually require custom designs. Extremely complex endeavors typically need special support from both Oracle and IBM. Nevertheless, a few properties of these types of architectures are worth mentioning.

Publicly accessible enterprises usually require a secure logical architecture, but are modified to provide continuous availability. See 3.3.4, "Very high internal security" on page 34".

Opening internal systems to the Internet immediately mandates the highest levels of security and generally implies 7x24 hours of operation. Edge Servers can be deployed into strategic areas of the Internet to speed access to static Web content, although the EnterpriseOne application itself is often constrained by the performance of the back-end and not the speed that static content can be delivered. Still, the benefit of forward locating this content can only be determined by an analysis of the anticipated volumes and types of Web information to be served. This can vary a great deal from client to client.

The high volume environments that are endemic to publicly accessible enterprises can be scaled up either horizontally or vertically. *Horizontal scaling* involves adding more physical machines. *Vertical scaling* involves adding capacity to an existing physical machine. PC environments tend to use horizontal, rather than vertical, scaling because benchmarks demonstrate rapidly diminishing returns by adding additional CPUs. UNIX and iSeries machines, which more efficiently support additional CPUs and can take more memory, tend to use vertical scaling.

Generally the best practice is to use vertical scaling because of its efficiency. Classes are loaded fewer times. There is more run-time reuse of components. System management is easier and hardware costs are reduced. Plus deployment is simplified as a result of having fewer containers. However, this practice must be weighed against the vertical scaling efficiencies of the underlying operating system.

#### 5.5.4 Running additional applications on the hardened J2EE application server

Portal, the Collaborative Server, and the EnterpriseOne Web Server are ultimately J2EE applications that run in the WebSphere Application Server. They are also all EAR files that are deployed to the application server environment and run on one or more JVMs. However, the server is also capable of running other J2EE applications. The WebSphere Application Server offers all the normal advantages associated with a J2EE environment, including fast, secure, and reliable server-side applications founded on an easily integrated and extended component model.

When adding additional J2EE applications, it is important to understand the additional demands on the application server to size the hardware correctly. The IBM Technology Foundation sizing questionnaire, available from your IBM hardware vendor, guides clients through quantifying the increased demand and determines the required hardware.

#### 5.5.5 Extending the architecture using Java Message Service

It is common for large corporations to integrate their applications that run on numerous hardware platforms with WebSphere MQ messaging products.

**Note:** MQ may not be included in Technology Foundation, and the Java Messaging Service (JMS) engine may require a separate purchase. Check your licensing agreement for details.

Technology Foundation can be augmented with a *messaging engine* so that its WebSphere applications can both integrate with JD Edwards applications and with the vast number of other corporate applications on the underlying MQ communication network.

You can also use non-MQ-based messaging products for the underlying layer. However, WebSphere does not support many essential services such as connection pooling for external vendors. Moreover, WebSphere only supports external vendors who support the Application Server Facilities portion of the JMS standard specification.

Messages can be synchronously or asynchronously delivered both through point-to-point or publish and subscribe methods in a secure, fault tolerant, workload sensitive, rules-based environment with sophisticated error handling.

**Note:** WebSphere fully supports the new EJB 2.0 specification for Message Driven Beans (MDB) that provides true asynchronous messaging.



WebSphere provides this light integration capability by supporting the industry standard JMS interface. While some clients may not immediately possess the required skill sets to enhance their Technology Foundation environment in this manner, IBM offers training in these techniques. The pay off for integrating a large number of interdependent but electronically isolated corporate systems can be enormous.

### 5.5.6 Using WebSphere Enterprise Edition features to do business process modeling

Since Technology Foundation customers are permitted to use any version of WebSphere, they can enjoy the benefit of the process modeling capabilities of the WebSphere Enterprise Edition known as *Process Choreographer*.

**Note:** Process Choreography also comes with the Integration Edition (IE) and the Enterprise Developer versions of the WebSphere Application Developer offerings.

IBM offers another process modeling product outside of Technology Foundation, called *WebSphere Business Integration Workflow* (formerly known as WebSphere MQ Workflow). WebSphere Business Integration Workflow is not tightly coupled with J2EE software as is Process Choreographer. However, it offers other advantages such as a C++ programming interface. For a comparison of these two products, along with an analysis of when to use which product, refer to the “Comparison with WebSphere MQ Workflow” section in *WebSphere Application Server Enterprise V5 and Programming Model Extensions: WebSphere Handbook Series*, SG24-6932.

Think of Process Choreographer as a highly visual tool that allows businesses to assemble new software rapidly by using portions of existing software. A new “flow” can be assembled (or an existing one modified) easily from software assets such as EJB method calls, J2C Connector Architecture (JCA) interfaces, and published Web services. Process Choreographer provides the ability to model manual human processes in the software such as having to obtain a manager’s signature. Assembling or altering the order of events in a software process becomes trivial through the Process Choreographer graphical user interface (GUI). Process Choreographer can make code changes completely unnecessary when the business evolves in a way that requires simple changes to existing software work flows.

Process choreography is founded on separating business functions from the order of execution of software assets. For example, the business function of marking an order as “booked” is separated from the work flow processes of having that order manually approved, ordering the associated bill of materials perhaps tracked in a separate product management system, and allocating shipping resources in yet another system. As software systems mature, code changes are more often caused by workflow changes than business function changes. Process Choreographer is a programming tool that has a simple user interface to provide developers with the ability to diagram new work flows using the full suite of the client’s business functions, regardless of where the software for that function resides or on which platform it runs. Code formerly available only to mainframe processes, for example, can now be made available to any approved enterprise application.

When the work flow is created (or existing work flow is modified), Process Choreographer automatically generates the associated software, ready for testing and deployment. Manual processes, such as waiting for a manager to physically sign and authorize some sort of action, are modeled as transactions that may or may not succeed.

**Note:** In J2EE terms, EJBs are produced and deployed to the WebSphere Application Server.

One way that Process Choreographer affords access to business functions is by providing a Web services interface to that function. This approach has the beneficial side effect of creating clear, Web-based interfaces to internal systems that can be used to support integration with external businesses. True business-to-business communication becomes possible as a fortunate by-product of Process Choreographer's use of Web services.

Since process modeling requires some J2EE expertise, many EnterpriseOne clients will not be interested or sufficiently skilled in this area. However, Process Choreographer can have a powerful and positive impact particularly for clients with rapidly evolving business environments by reducing software development time, integrating disparate software systems on different hardware platforms, parallelizing tasks that were formerly executed linearly, and increasing business to business support. Moreover, IBM offers demonstrations and training in Process Choreographer for clients who are frustrated by slow software systems that always seem to lag behind the requirements of their business.

## Sample logical architecture selection document

This appendix shows a sample logical architecture proposal document to be submitted to the end client by the hardware vendor. Notes to the document author, presumably the hardware vendor, are shown in shaded boxes.

This sample document has been prepared for the fictitious company called “Acme Widgets”.

# **Proposed High Availability Architecture for Acme Widgets, Inc.**

**Version 1.0**



## Table of contents

**Note to hardware vendor author:** The table of contents should present a clear outline of the architecture proposal to the client

Executive summary .....	88
Business overview .....	88
Business objectives .....	89
Acme's current environment .....	90
Current physical architecture .....	90
Current user community .....	91
Current security mechanisms .....	92
Recommended logical architectures .....	92
Initial rollout recommendation .....	93
Medium-term recommendation .....	95
Long-term recommendation .....	96
Rationale for architecture proposal .....	98
Required skills .....	99
Next steps .....	101
Sample proposal glossary .....	102

## Executive summary

This document proposes a logical architecture for a high availability Web tier for the Acme Widget deployment of JD Edwards' EnterpriseOne application. The document outlines the immediate and long-term business objectives associated with deploying the application. It describes Acme's current Enterprise Resource Planning (ERP) user community and technical environment. Then it proposes various logical architectures according to anticipated phases of growth.

The initial rollout architecture supports high availability, fault tolerance, and an improved security model. At Acme's request, subsequent sections show additional logical architectures capable of supporting more users and providing increased security.

Key assumptions are documented here. It is imperative that Acme review these and verify their correctness before proceeding with the first implementation steps.

This document comprises the first step of a multi-step, iterative process to create a fully defined physical architecture for Technology Foundation. Hardware sizing, the next step, is not part of this document.

**Note to hardware vendor author:** The executive summary is the only section that managers often read, so tailor the material for them. Tell them what the project is about and what you expect of their people.

## Business overview

"Incorporated in 1921, ACME WIDGETS builds widgets and other infrastructure-related objects used by millions of people. In addition, Acme produces trinkets, bobbles, ornaments and doodads and other Widget materials. Unusual among large widget producers, Acme handles both large and small production through its three operating divisions. The Branch Division serves local markets in the West and builds smaller, quick fix widgets. The Heavy Widgets Division (HWD) builds larger, more complex widgets nationwide.

The group's principal activity is to provide widget materials and services. It constructs infrastructure widget materials like trinkets, bobbles, and doodads. The group also performs widget preparation services for buildings, plants, private and government institutions.

The operations are carried out through the three divisions: the Branch Division, the Heavy Widgets Division, and the Operational Services Division. The Branch Division is comprised of 13 sub-branches that serve local markets and includes Widget infrastructure and the improvement and design of new widget components. The Heavy Widgets Division pursues major infrastructure widget establishment throughout the nation and includes activities such as demarcation, dusting, de-widgetizing, and prepping. As of 30 April 2002, the company acquired the Tougher Widgets Company.

The branch division accounted for 63% of 2002 revenues, the Heavy Widgets Division, 24%, and the Operational Services Division produced 13%", according to SmartMoney.com. Acme has net sales of \$1,237,456,789 (as of 31 December 31 2002) and its ticker symbol is "WGT".

**Note to hardware vendor author:** Demonstrate that you have a basic understanding of the client's business. Embarrassing mistakes, often at a gross level, can be avoided by ensuring that all team members (especially "techies") comprehend the high-level business fundamentals.

## Business objectives

Acme has divided their objectives into immediate, long-term, and very long-term categories. The immediate objectives are:

- ▶ To deploy a stable, reliable, portal-enabled Xe and ERP 8.x compatible version of EnterpriseOne that is capable of supporting 500 concurrently logged in users
- ▶ To ensure that the ERP environment supports high availability  
Continuous availability is not necessary since planned downtimes are acceptable during weekends and off-peak hours.
- ▶ To integrate the EnterpriseOne security mechanisms with Acme's existing corporate Active Directory servers
- ▶ To support open enrollment by October 2004 (previously October 2003)
- ▶ To customize the look and feel of the Web pages for Acme
- ▶ To tailor the portal environment with custom portlets

A few examples of these potential new capabilities include:

- Access to e-mail
  - Job cost estimation
  - e-learning
  - Checking the weather for Widget production work
  - Providing maps for shipping project materials (both interstate and local)
  - Instant messaging
  - Chat and online conferencing
- ▶ Portal potentially will become the new online workplace. Acme is uncertain which, if any, of the portlets will be used. The addition of more portlets may begin as early as March 2003.  
Acme-proprietary portlets may be developed by contracting out the work, but Acme does not intend to develop their own portlets until they address their very long-term objectives.
- ▶ To provide different portal views based on role (project manager, administration, etc.)
- ▶ To perform security administration for all applications from a single place  
For example, the Lightweight Directory Access Protocol (LDAP) and ERP security information can be managed from a single administration interface.
- ▶ To Allow end users to customize their own Portal environment (personalization encourages adoption)
- ▶ To establish secure socket connections (Secure Hypertext Transfer Protocol (HTTPS)) for employees to sign up for benefits from home  
Future plans include making the "self-service" applications available on the Internet and accessible outside the firewall.

**Note to hardware vendor author:** Users have a tendency to want to categorize all requirements as short-term and of the highest priority. Part of the architect's job is to ensure that the project goals are technically and practically achievable. If users are reluctant to think in terms of trade-offs, numbering the requirements in the order of their importance is a technique that can help force priorities.

The long-term objectives are:

- To develop custom portlets to support, for example, the ability to purchase and sell employee vacation days

Establish an environment from development groups, an integration environment for those groups, a quality assurance environment for testers, and a distribution mechanism for the custom portlets. Develop methods and practices to support custom development efforts.

- The user community is ultimately expected to grow well beyond the current 500 ERP users. The system must support greater simultaneous load.

## Acme's current environment

The following sections describe Acme's current physical environment, user community, and security mechanisms.

**Note to hardware vendor author:** The client will respect your recommendation only if you demonstrate that you have a good understanding of their technical environment.

### Current physical architecture

Figure A-1 shows Acme's physical architecture as it exists today.

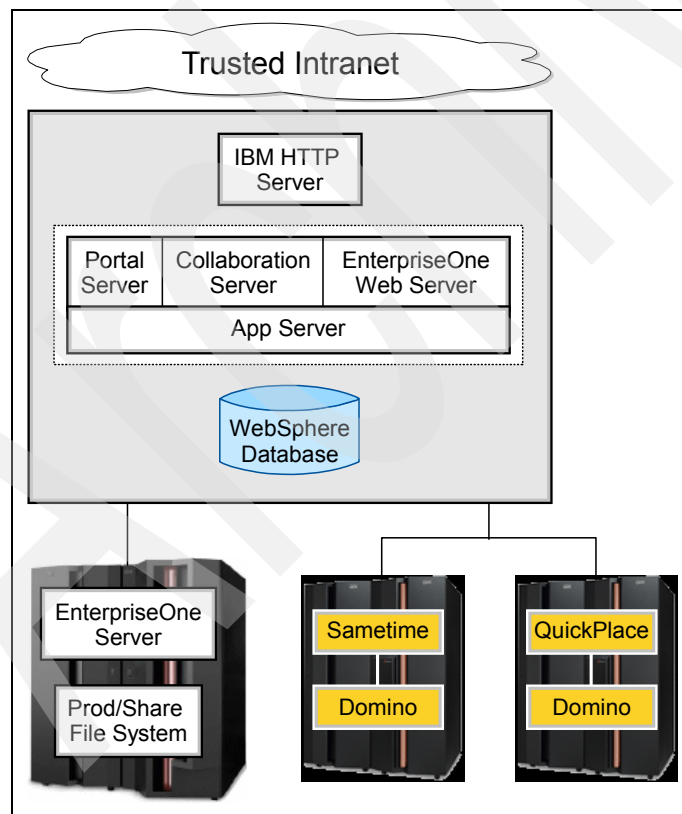


Figure A-1 Acme's physical architecture today



Note the following observations and assumptions:

- ▶ The gray boxes represent PCs while the EnterpriseOne server is an iSeries.
- ▶ The WebSphere database collectively refers to the WebSphere Application Server, WebSphere Member Services, and WebSphere Portal Server databases.
- ▶ Lotus Instant Messaging and Web Conferencing (Sametime) and Lotus Team Workplace (QuickPlace) are not currently used in production.
- ▶ Active Directory provides Acme's corporate LDAP information and is configured as a high availability server. However, the interface to the corporate LDAP does not yet exist. Domino is currently being used as the LDAP server. The instance of the Domino server eventually will be used only to support Collaborative Portal capabilities. It is not intended to be a permanent LDAP server or to support the EnterpriseOne components in any way.
- ▶ Acme has a preference for IBM PC hardware due to the direct binding and tighter coupling to iSeries hardware by xSeries PCs.
- ▶ Acme uses an iSeries Model 870 with 16 GB memory, 1 TB for storage, and 8 processors.
- ▶ The Windows machines currently have two CPUs.

## Current user community

The following attributes characterize the current user community:

- ▶ There are 1800 potential users.
- ▶ All users are Acme employees. Access to EnterpriseOne for companies or individuals outside of Acme is not required.
- ▶ At most, 500 users are concurrently logged on (limit imposed by EnterpriseOne license provides this upper bound).
- ▶ OneWorld users are geographically distributed. For example, administration and accounting is done either at the headquarters in New York or at the branch offices while services are performed in the field by 1000 to 1500 project managers. In each of the local offices, employees enter timecards, accounts payable vouchers, etc.

These attributes characterize the application environment for the current user community:

- ▶ Other than the corporate Active Directory service, Acme states that they have no new EnterpriseOne external applications to integrate with. Since all of their applications are EnterpriseOne applications, they are already integrated on the iSeries server.
- ▶ There is no short-term, immediate need to integrate with any EnterpriseOne-external applications through Technology Foundation components that may pertain to trinkets, design, Widget services, or shipping systems. For example, Portal's credential vault will not be used to integrate with EnterpriseOne external applications.
- ▶ Acme intends to use nearly all of the EnterpriseOne applications immediately including time-entry, self-service, address-book, etc. They intend to use everything except the customer relationship management (CRM) capabilities.

## Current security mechanisms

The following attributes characterize the current security environment:

- ▶ All users access the ERP environment from within a trusted, private Acme network. Branch offices are connected with private frame-relay lines. Internet access to the private network is secured with virtual private network (VPN) software using digital certificates. Also, a few dial-up connections to the private network exist.

- ▶ The Acme private network is linked to the Internet through a firewall-demilitarized zone (DMZ)-firewall barrier. No reverse proxy elements or Web tier authentication capabilities are in place at this time.
- ▶ All Windows machines have Norton Antivirus and receive periodic virus updates.
- ▶ All elements inside the Acme network are completely trusted.
- ▶ No firewalls are in place to protect the Technology Foundation PCs from Acme's other corporate PCs.

## Recommended logical architectures

The following sections show the initial and immediate recommendation, followed by recommendations for medium and long term growth. "Rationale for architecture proposal" on page 101 describes the justification of logical architecture selections.

**Note to hardware vendor author:** Divide the architectures according to stages of a long-term initiative. Frequently the hardware platform significantly shifts as the user community grows, often away from a small, PC-based architecture toward a more reliable platform. Have the client think about what shifts will be ultimately necessary early on.

### Initial rollout recommendation

Figure A-2 shows the recommended logical architecture.

**Note to hardware vendor author:** There are two customizations to this Standard Security, Highly Available logical architecture. There is an external LDAP server, and the availability for the non-mission critical Lotus elements has been degraded.

Note the following observations and key assumptions:

- ▶ Lotus Instant Messaging and Web Conferencing and Lotus Team Workplace are assumed to be non-mission critical components for the rollout architecture and, therefore, do not provide high availability.
- ▶ Dotted lines represent separate pieces of physical hardware. For example, the dotted line around the EnterpriseOne server represents the iSeries box. For Acme, the other dotted line boxes are likely to be mapped to Intel boxes. This mapping occurs during the sizing process that follows this logical architecture definition.
- ▶ The fault tolerance for the EnterpriseOne server itself is out of scope. This document addresses only Technology Foundation components. EnterpriseOne servers, by default, provide fault tolerance at the process level at the very least.
- ▶ Acme has a pre-existing external directory server (Active Directory). The configuration of this external server, for example to provide high availability, is out of the scope of this project although integration to the services is not.
- ▶ Unlike the growth architecture described in the following section, this topology cannot withstand a failure during a system upgrade.
- ▶ This architecture requires users to re-login if their associated WebSphere Application Server machine fails. In-flight transactions are lost in the event of failover. However, users can immediately re-login while the failed WebSphere Application Server machine is fixed.
- ▶ Firewalls are added to protect against corruption to the inner network (the vast majority of corporate virus attacks originate from within the trusted network). While these

components do not represent the highest level of security, which involves additional internal barriers, the architecture constitutes a vast improvement over the current topology.

- ▶ This architecture can support limited non-fault tolerant or experimental requests from:
  - E-learning
  - Chat
  - Lotus Instant Messaging and Web Conferencing messages
  - Lotus Instant Messaging and Web Conferencing responses
  - E-mail
  - Map requests
  - Weather requests
- ▶ Portal sizing, which occurs during the hardware ordering process, determines the number of portal servers necessary to support the anticipated portal user interface rendering load.
- ▶ An identical quality assurance environment is established to perform regression testing, performance load analysis, failover testing, and various other QA tests. Upgrades are only installed during planned downtimes. Changes to the production environment are properly tracked in a configuration management database. Defects are also tracked.
- ▶ For simplicity, the box labeled *WebSphere database* collectively refers to the WebSphere, Member Services, and Portal databases.

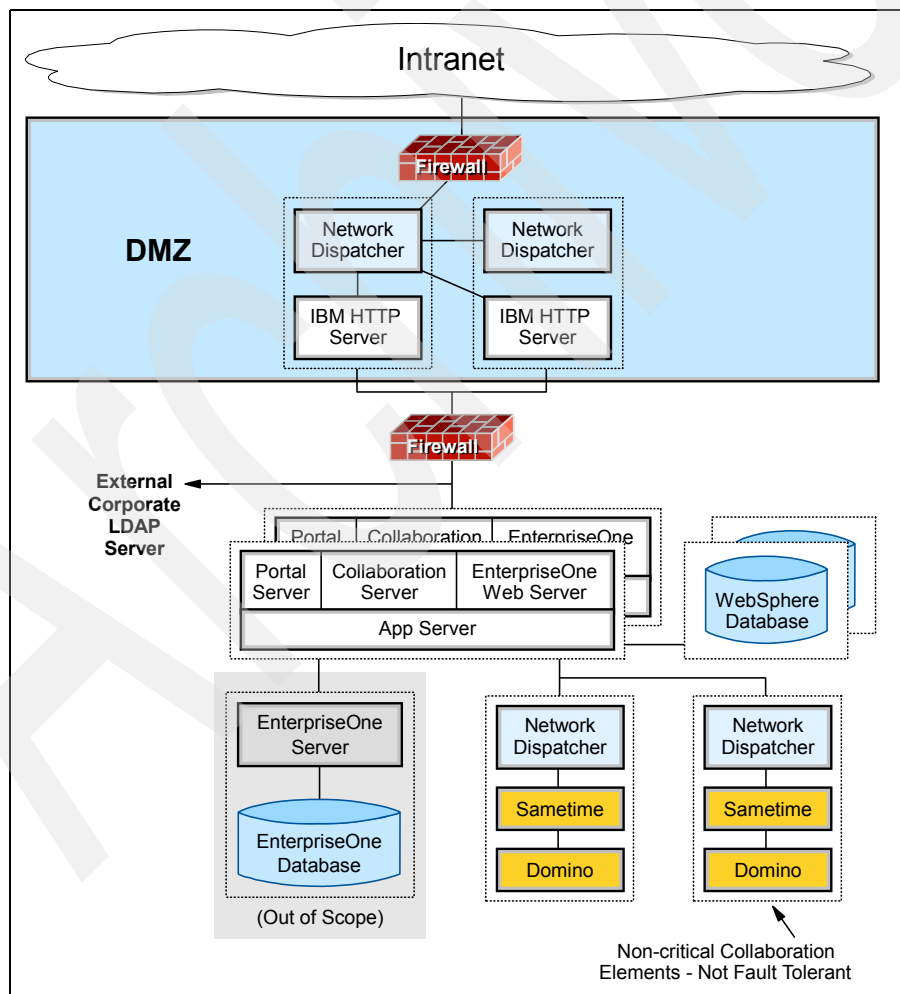


Figure A-2 Recommended logical architecture for Acme

In the diagram in Figure A-2, the software groupings are represented by a surrounding dotted line. This is intended to show that the group is likely to reside on the same physical machine (or logical partition, in the case of an iSeries). Each group of software components can be mapped into one of many supported hardware platforms during the sizing process. Some of the potential mappings are shown in Figure A-3.

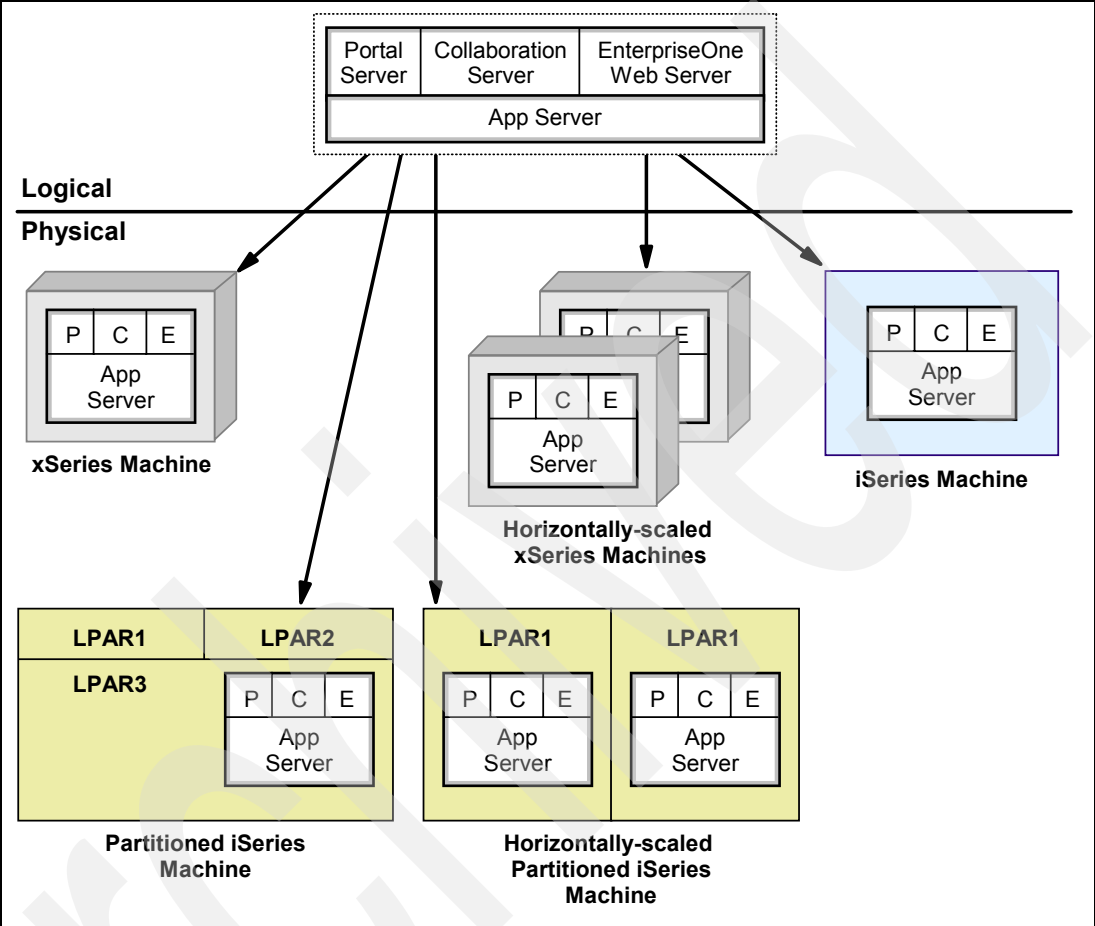


Figure A-3 Software component and hardware platform map

Figure A-3 illustrates how a single component grouping from the logical architectures represented in this document can map to an Intel machine, to an iSeries partition, to multiple horizontally scaled machines, or to other variations. For example, the WebSphere database may ultimately be mapped to the iSeries server. Because the number of potential mappings is virtually infinite, JD Edwards appropriately limits the supported choices. By limiting the supported end physical architectures, JD Edwards allows for all supported options to be thoroughly tested and debugged and improves quality and performance for end clients. The mappings from the logical architectures to the physical realizations are maintained internally by JD Edwardse and are part of the sizing process that follows the generation of this document.

**Note to hardware vendor author:** Tell the end client that a formal physical architecture proposal will be forthcoming, after the logical architecture is agreed to.

## Medium-term recommendation

Acme has a requirement that the system be architected in a scalable fashion. The logical architecture is intended to show how the recommended architecture can be horizontally expanded. The architecture scales by parallelizing work across multiple Web containers. Figure A-4 illustrates a recommended logical architecture for a medium-term solution.

Note the following observations:

- ▶ The Portal usage and its extended features have a particularly powerful impact on the sizing model for this architecture.
- ▶ Redundancy for Lotus Instant Messaging and Web Conferencing and Lotus Team Workplace requires Domino Enterprise, which is not currently included in the base Technology Foundation offering.

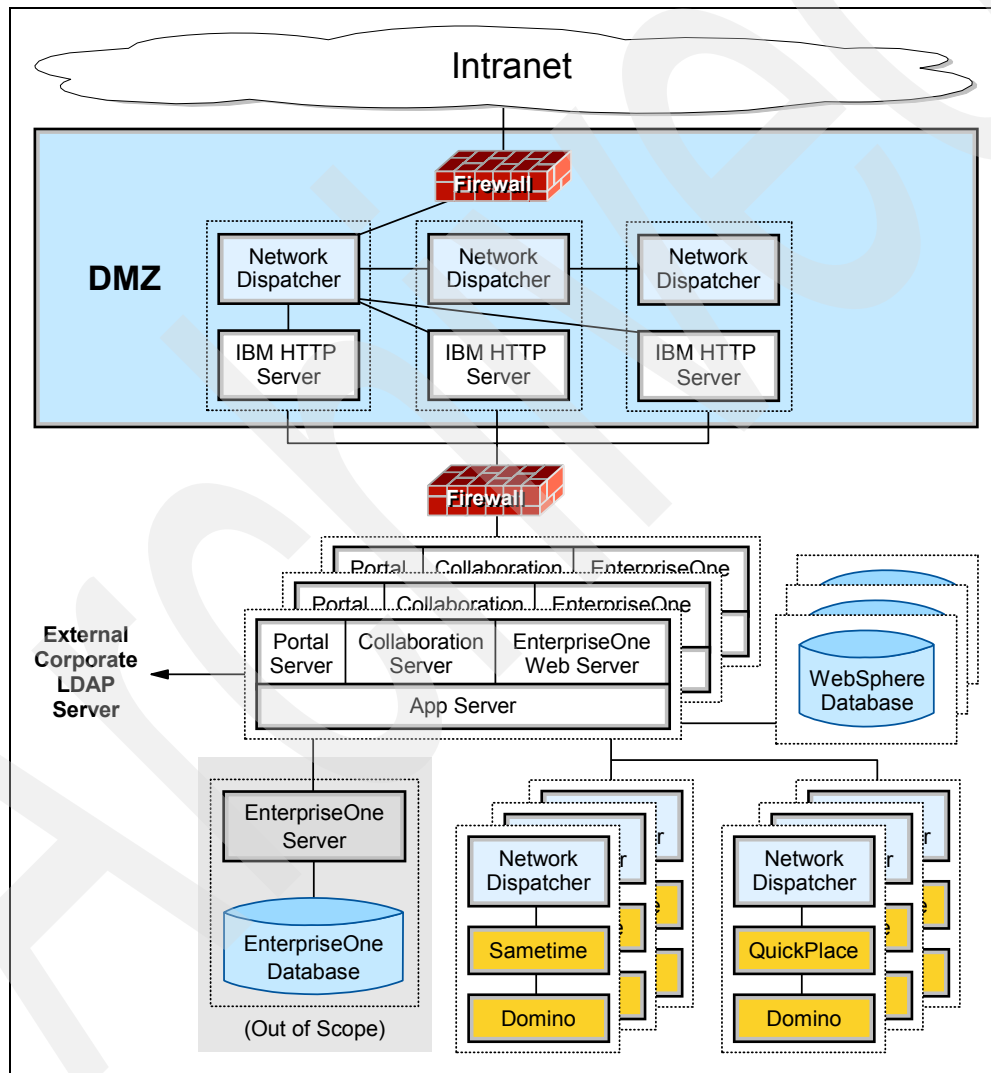


Figure A-4 Recommended medium-term logical architecture for Acme

## Long-term recommendation

Acme has further requested a proposed architecture that is capable of providing access to a much larger and diverse user community, possibly including non-employees. Figure A-5 illustrates the architecture offered in response to this requirement.

Note the following observations:

- ▶ Edge Servers are added to forward locate information closer to Acme's remote offices in the proposed locations of Canada and Florida. This provides profound performance improvements for those remote offices since they are anticipated to have slower T1 or integrated services digital network (ISDN) connections to the data center.
- ▶ WebSEAL components provide reverse proxy to improve security in an ever widening and less trusted internal network.
- ▶ Although the mapping to physical boxes is part of the sizing exercise, long-term throughput requirements likely require that the Technology Foundation components be moved to the iSeries server. The iSeries provides vastly improved performance over attempts to horizontally scale Intel platforms, which provides diminishing returns. However, running Lotus Instant Messaging and Web Conferencing and Lotus Team Workplace on the iSeries is not yet an officially supported by Oracle. This architecture is provided only to fulfill Acme's requirement of visualizing long term approaches.
- ▶ This architecture relies on Portal Express Plus.

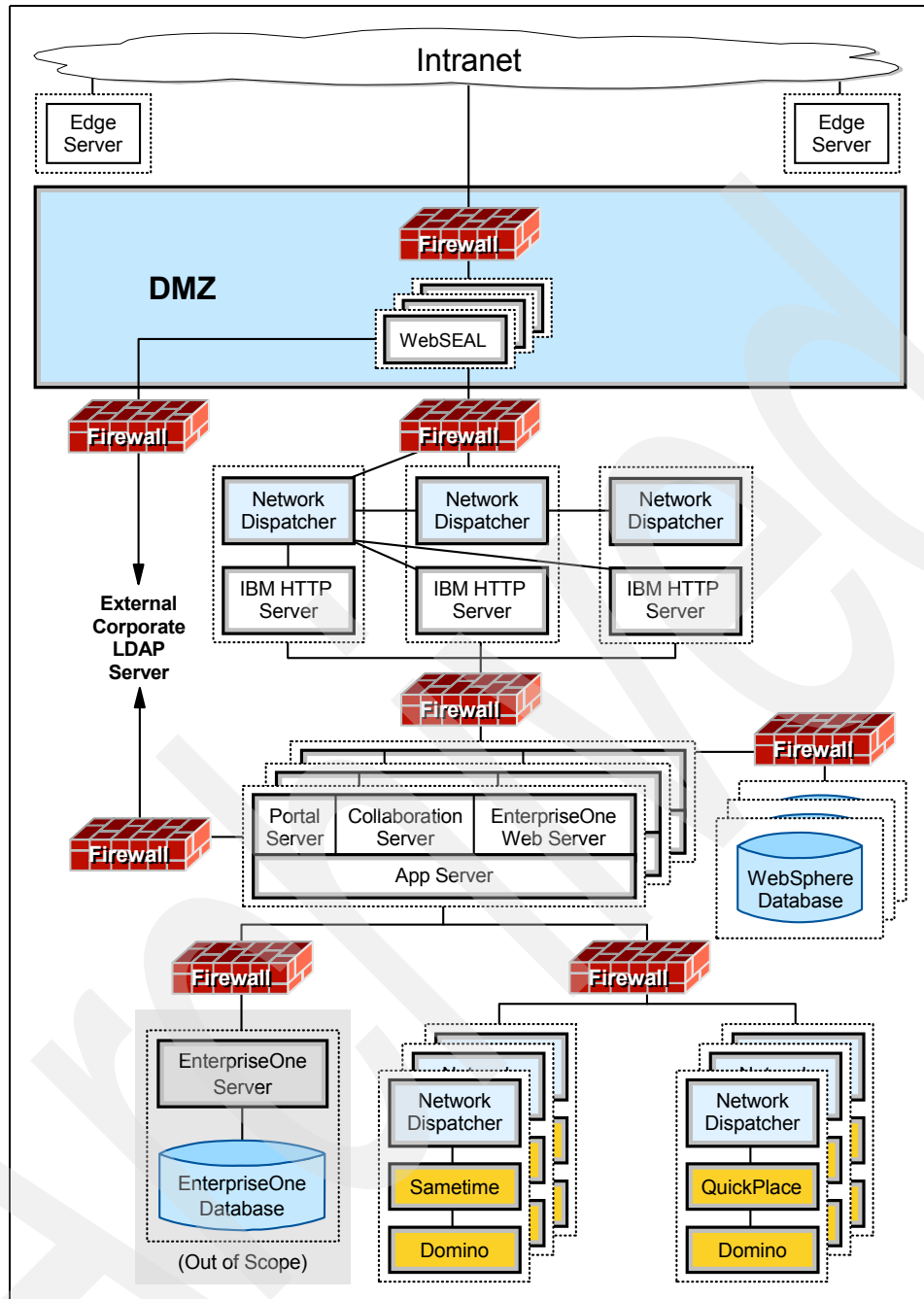


Figure A-5 Recommended long-term logical architecture for Acme

## Rationale for architecture proposal

The following decision factors were considered particularly key in the logical architecture selection:

- ▶ Availability requirements
- ▶ Security requirements
- ▶ Implementation time frame, existing Acme logical and physical architectures, and miscellaneous factors

The following sections address each of these factors in particular.

## Justification for availability

Interviews suggest that Acme's availability requirements are serious, but not "life threatening". Continuous 7x24 hours of operation is not a requirement. Planned downtimes, for example during slow hours on weekends, are acceptable. Non-functional requirements are significant, but the system is not classified as one that must stay operational at all costs. Moreover, performance is valued but does not warrant the most expensive systems available. Therefore the initial rollout recommendation does not provide continuous availability. The proposed initial rollout architecture is only horizontally scaled enough to support high, rather than continuous, availability.

## Justification for security

While Acme's present architecture is adequately protected from Internet security compromises, it is highly vulnerable to attacks from within (the vast majority of security compromises occur from within). While each PC is protected with antivirus software, such software often flags problems only after they have occurred. It is appropriate to provide additional protection to the enterprise software PCs as compared to the protection afforded to the average employee's desktop. The recommended architecture uses firewalls to segment off and protect the critical enterprise network from the remainder of the corporate network.

Acme's EnterpriseOne system and HTTP servers do not serve any information over the public Internet. Some moderately sensitive employee information such as vacation hours, medical plan selection (but not the personal medical information itself), hours worked, etc. is served over the corporate network. The EnterpriseOne information pertains to customer orders and distribution information. This information, if compromised to Acme employees on the internal network, causes relatively light damage to the business. Moreover, Acme has no corporate standard to use the highest security configurations. On the contrary, the business is only now adopting professional security standards.

## Miscellaneous

The initial rollout is scheduled to be completed by early 2004. The initial rollout topology must be simple and straightforward to be implemented in this time frame. The rollout architecture is easily configured and maintained.

As Acme completes this logical architecture selection process, it will proceed toward the sizing process, beginning with the selection of a physical platform. For Acme, the physical platform is likely to be determined by its hosting strategy roadmap. Acme already possesses in-house iSeries and Windows expertise, which likely eliminate a pSeries or UNIX approach. Since Oracle does not yet support Portal on iSeries, Windows machines are mandated for the Portal Server and Lotus Instant Messaging and Web Conferencing and Lotus Team Workplace components.

## Required skills

Table A-1 lists some of the skill sets that are required to install and maintain the proposed architecture. While a single person can certainly perform more than one role, the skill sets are distinct. A column is included to show whether the particular skill ordinarily needed to install and maintain Technology Foundation will be immediately required for Acme. For example, Acme has no production Portal requirements, so strong expertise ordinarily needed in many of the Portal-related tasks is not required.



Table A-1 Required skills to install and maintain the proposed architecture

Role	Skill	Skill required for Acme?	Reason
WebSphere administrator	Basic WebSphere Application Server administration.	Yes	Technology Foundation uses WebSphere. Administrator must be able to establish clones and server groups, understand and alter XML configuration files, institute runtime monitoring, understand resource analyzer, perform problem isolation for application servers, and deploy EAR files, install patches.
	WebSphere Application Server workload and cluster management expertise	Yes	Recommended topology is high availability.
	WebSphere Application Server Network Deployment Management Expertise	No	Acme is not on WebSphere Application Server Version 5.
	WebSphere Application Server tuning expertise	Yes	Administrator must know how to configure threads, connection backlogs, servlet caching, etc.
	Basic HTTP Administration	Yes	Administrator must understand how HTTP plug-ins work to link WebSphere Application Server with the HTTP server.
	HTTP Tuning expertise	Some	Administrator must understand how to configure Apache thread daemon configuration.
Database administrator	DB2 tuning expertise	Some	Presumably Acme's back-end is already tuned and has a database administrator. For Technology Foundation, there may be some additional WebSphere Application Server database tuning.
Network and security administrator	Basic network administration	Yes	Configure Technology Foundation machines, Domain Name System (DNS), TCP/IP.
	Firewalls, threat detection, reverse proxy, threat analysis	Yes	Administrator must understand exactly how to configure the firewalls. Any mistakes in this domain can expose the system to security threats.
	Authentication mechanisms and authorization models	Yes	Administrator must understand how Domino LDAP is used. Must know how to convert to Active Directory LDAP (Acme corporate standard).
Portal developer	Portal Development	No	Acme has no immediate plans to develop their own portlets. Any short- or medium-term portlet development will be contracted out.

Role	Skill	Skill required for Acme?	Reason
Portal administrator	Portal Administration and Access Control	Yes	Administrator must understand basic Web modules and WAR files, deployment techniques, portal security (user and group management), scripting, and performance monitoring.
	Portal Web Clipping, Themes and Skins, Cascading Style Sheets	Yes	Acme has short-term plans to customize the look and feel of their Web pages.
	Portal Failover	No	Administrator must understand Portal failover particularly as it pertains to Domino, Lotus Instant Messaging and Web Conferencing, and Lotus Team Workplace and using clustering in conjunction with CMP and Network dispatcher to achieve failover. This is not yet required since Acme's initial- and medium-term plans do not require Lotus Instant Messaging and Web Conferencing and Lotus Team Workplace to be treated as mission-critical elements.
Lotus Collaboration administrator	Lotus Instant Messaging and Web Conferencing	Some	Administrator must establish instant messaging and person awareness. Only moderate expertise is required because Acme has short term plans to evaluate the usefulness of this product.
	Lotus Team Workplace	Some	Establishing a collaborative work area. Only moderate expertise is required because Acme has short-term plans to evaluate the usefulness of this product
	Lotus Domino	Yes	Administrator must understand LDAP authentication and performance monitoring techniques. Acme short-term plans require using Domino as the LDAP server.
System administrator	Administrator capable of understanding system as a whole. Able to isolate problems that span tiers, recommend physical topology changes to address performance problems, able to institute cross-tier performance monitoring mechanisms.	Yes	Problems cross application boundaries and physical machines.

The project should be staffed to accommodate both planned and unplanned absences as well as normal attrition rates.

**Note to hardware vendor author:** Frequently, clients have unrealistic expectations about the necessary efforts behind a true enterprise application that serves hundreds or thousands of users. This section is about setting realistic expectations, which is one of the keys to successful project management. Surprisingly, even multi-billion dollar companies envision a single, untrained system administrator as having all the required skills for WebSphere administration and tuning, Portal configuration, security, database tuning, network administration, and so on. While the installation and maintenance of Technology Foundation is well-document, and some administrators are truly gifted with multiple talents, it is still a good idea to explicitly list the required skills to avoid ultimate disappointment.

## Next steps

The potential action items listed in Table A-2 remain.

*Table A-2 Potential action items*

Potential action item	Approximate time frame	Responsible party
Set up meeting to gather requirements for portlets with human resources (HR) staff responsible for employee self-service capabilities.	3 February 2005	Acme – John Smith
Demo of personalization and collaboration elements to Acme.	15 January 2005	Oracle/IBM – Jane Smith
Submit architecture for formal hardware ordering and precise sizing (memory requirements, disk drive configuration, etc.)	15 February 2005	Acme and Acme's chosen hardware vendor who is supported by IBM sizing experts on the back-end

**Note to hardware vendor author:** The information in this section is essential for maintaining project momentum. Assign specific names and dates to action items.

## Sample proposal glossary

The glossary often ends up as a key mechanism for driving out differences in language use. Be sure to include client-specific term. If they have their own commonly used terms, define them and demonstrate an understanding of their business. A well-written glossary can easily be converted into training material for new project members as well.

**Note to hardware vendor author:** Review each definition in the client's architecture proposal glossary to ensure the definition is appropriate for the client's technological level of understanding. Remember that the intent is to add understanding and arrive at a meeting of the minds rather than to overwhelm the client with technical jargon or to attempt to impress them with sheer volume of information. The goal is clarity.

Add a custom reference section if necessary.

### **conference room pilot (CRP)**

Acme's term for a prototyping environment.

### **virtual private network (VPN)**

A private communications network used within a company, or by several different companies or organizations, to privately communicate over a public network like the Internet. VPNs provide Acme employees secured access to its private network when they log in from home.

**Note to hardware vendor author:** This definition is simplified and customized for Acme. A lot can be said about VPNs, but for this client, only the information they really need is included.

**Note to hardware vendor author:** Insert your standard disclaimer page here.

© Copyright IBM Corporation 2004

This document was prepared by IBM in conjunction with Oracle and Acme Widget. The recommended architecture is an approximation of the resources required to support Acme's ERP implementation. It is an effort based on information available at a point in time, and gathered over the course of a number of phone conversations. The customers' actual requirements may vary from the estimated requirements because of unanticipated work loads, new vendor relationships, emerging requirements, and numerous other factors.

IBM, the IBM logo, AS/400®, DB2, DB2 Universal Database, Enterprise Storage Server®, @server, iSeries, Netfinity®, pSeries, RS/6000®, Tivoli, WebSphere, and xSeries are trademarks or registered trademarks of IBM Corporation in the United States, other countries, or both.

UNIX is a registered trademark of the Open Group in the United States and other countries.

Widget is a registered trademark of the Acme Widget Company.

Intel and Pentium® are trademarks of Intel Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems™, Inc. in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates. IBM hardware products are manufactured from new parts, or new and used parts. Regardless, our warranty terms apply.

©Copyright Oracle 2003  
Oracle  
One Technology Way  
Denver, CO 80237  
U.S.A.

The materials contained herein are summary in nature, subject to change, and intended for general information only. The materials reflect current plans for future software or enhancements that may require additional license fees to obtain, and are not a commitment of Oracle to develop or deliver such software or enhancements.

Oracle is a registered trademark of Oracle Corporation. The names of all products and services of Oracle used herein are trademarks or registered trademarks of Oracle Corporation.

U.S. or Canadian patents and patent applications may cover inventions used in the production of EnterpriseOne.

Other company, product and service names may be trademarks or service marks of others.

Archived

# Glossary

**continuous availability** A type of high availability involving three or more machines configured for redundancy so that, in the event that one of the machines fails, another machine takes over its workload. These three or more machines allow for additional failover protection. If only two machines, rather than three machines, are used to provide high availability, the system is vulnerable during upgrades. If one machine is in the process of being upgraded and is unavailable, the other machine must service all requests. During the interval when the first machine is serviced, a failure of the second machine means the system becomes unavailable. This is why three or more machines are necessary to provide *continuous availability*. Even a three-machine configuration is still vulnerable. Availability is improved by adding a fourth, fifth, or sixth machine. The degree of redundancy must be determined by comparing the costs of additional hardware, complexity, and maintenance to the cost and likelihood of the enterprise suffering downtime.

**denial of service attack** Effectively causes a Web site or network service to appear as down. There are many types of denial of service attacks, but most operate on the same principle. That is, they flood the site with so many requests that the site becomes overwhelmed and cannot service the requests or any other legitimate requests. Denial of service attacks exhibit the same symptoms that viruses do, since the system becomes unavailable. However, the similarity ends there, since the system is not infected with any foreign software and technically is fully operational. For more information, see:

[http://www.cert.org/tech\\_tips/denial\\_of\\_service.html](http://www.cert.org/tech_tips/denial_of_service.html)

**demilitarized zone (DMZ)** A publicly accessible area of a company's network that provides services to the Internet while protecting the company private network. Machines in the DMZ are moderately protected from the Internet, but less than the company's private network. Firewalls reside on both the internal and external sides of the DMZ. Typically, limited Internet traffic is allowed into the DMZ, but only company owned DMZ machines are allowed limited communication with the private network.

**Enterprise JavaBeans (EJB)** A Java application programming interface (API) developed by Sun Microsystems, IBM, and others. It defines a component architecture for multi-tier client/server systems. EJB systems allow developers to focus on the actual business logic, rather than worry about endless amounts of programming and coding tasks that are common to all pieces of software. Developers design (or purchase) the necessary EJB components and connect them together on the server. Because EJB systems are written in Java, they are platform independent.

**firewall** A hardware device that examines every packet that is passed between networks. Each packet is examined. Then it is either passed on or completely dropped according to the policy established by the security administrator.

**forward proxy** See *proxy*.

**high availability** Architectures that are capable of providing service even though some of their components can suffer failure. It requires at least two identical pieces of server hardware to provide failover. One physical server substitutes for the other when a failure occurs. During normal operation, one server can share the work to lighten the load of the primary server. However, this workload management is not essential to satisfy the definition of a highly available system. Architectures can also be partly highly available in that they can provide failover for components that are more likely to incur downtime, but provide no failover for components deemed more reliable, such as hardware firewalls.

**horizontal scaling** Typically used to allow more end users rather than to increase the response time for any single user (see *vertical scaling*). Horizontal scaling increases the number of physical machines.

**independent software vendor (ISV)** Produce generalized solutions rather than point solutions for a single end client. Oracle is an ISV since it produces software for end users.

**IP spraying** The piece of software (or hardware) that distributes packets destined for the same IP address across multiple machines, but with affinity that ensures one user's packets are usually routed to the same machine.

**JavaServer Page (JSP™)** A freely available specification to extend the Java servlet API to generate dynamic Web pages on a Web server. JSPs are based on a Hypertext Markup Language (HTML)-like language that is turned into a servlet. They allow people who are very familiar with HTML and graphics design to insert small portions of Java programming logic into their HTML. This allows Web page designers, who are not programmers, to execute logic inside of their HTML such as performing database requests or conditionally generating different parts of a Web page.

**Mpps** (Million Packets per Second) A common unit of measure for firewall efficiency.

**Javadoc** A tool that parses the declarations and documentation comments in a set of source files and produces a set of HTML pages describing the classes, inner classes, interfaces, constructors, methods, and fields.

**keepalive** A connection that remains open even after the request is serviced. The TCP connection is not automatically closed. When a keepalive connection times out, it should not be noticeable to the end user. No error is generated since it's not considered an error. The only way to notice the timeout is that, when another request arrives, there is a small amount of overhead setting up a new connection.

EnterpriseOne installations are not bottlenecked waiting for the delivery of static data. Consequently they are unlikely to observe noticeable performance improvement from keepalive configuration.

Two keepalive connections in the EnterpriseOne topology are of interest. One connection is between the browser and the HTTP server. The other connection is between the HTTP server and WebSphere. Experiments show that for browser-Apache and Apache-WebSphere connections, the default is to establish all connections as keepalive connections. Both the browser to HTTP server and the HTTP server to WebSphere connections are configured completely independently of each other and do not affect each other. One may be completely turned off, and the other would not notice.

HTTP was improved in version 1.1 to support persistent TCP/IP connections. For more information, see:

<http://httpd.apache.org/docs/keepalive.html>

**logical partition (LPAR)** The practice of dividing a single physical computer's resources (memory, disk, CPU, and resources such as the network card) into multiple logical machines. Each partition is managed by a separate copy of the operating system to help isolate segments of enterprise systems that are characterized by related pieces of software. Partitioning greatly aids in performance tuning and in debugging run time problems. A downside is that a single physical machine is not protected against power outages or network card problems. Both the iSeries (with OS/400) and the pSeries (with AIX UNIX) can be partitioned. The xSeries running Windows currently cannot be partitioned.

**Network Address Translation (NAT)** A component that sits between a company's internal network and the Internet. It translates IP address so that inbound packets are routed to the correct machine and outbound packets appear to be from a different IP address. Since NAT hides a company's true internal IP addresses, it increases security. It also allows a company to use any IP address even if it conflicts with addresses on the Internet since, before the packets leave the internal network, they are translated into an approved address. This increases the available IP space inside of the trusted network.

**Netwm** An EnterpriseOne utility capable of identifying the current state of the workload of all kernel processes. Netwm is useful to determine whether a single EnterpriseOne kernel has a backlog of requests in its queue. Consult your EnterpriseOne documentation.

**proxy** A machine that hides the identity of a trusted internal machine to untrusted external machines. As the trusted machine attempts to access information from external, untrusted machines, the proxy acts as an intermediary and substitutes its own network address for the requester's address in network packets before sending the packets out. The untrusted machine returns the information to the proxy and the proxy returns the information to the trusted, internal machine. The proxy essentially acts on behalf of the requestor.

**reverse proxy** Behaves essentially the same way as a proxy except the external, untrusted machines originate the request. Reverse proxies are used to conceal the existence of internal machines, such as HTTP servers, from untrusted machines on the Internet. To the untrusted machines, the reverse proxy machine appears to be the HTTP server. Since the difference between a proxy and a reverse proxy is subtle, and the same software and machine can act as both, the terms are often confused.

**pstack** An AIX utility that shows the current program stack for any UNIX process. When pstack is used against EnterpriseOne kernel processes, it is possible to identify the function call in which the kernel is currently located. Consequently, it is possible for long running SQL calls to determine whether the kernel is bottlenecked waiting on the database. See your AIX documentation.



**servlet** A small lightweight Java program executed on the WebSphere Application Server used to support HTTP requests by dynamically and programmatically assembling HTML. Servlets are needed because static HTML simply cannot dynamically assemble a Web page that may, for example, require information from the database. When executed, servlets typically stay in memory so they do not have to be restarted for the next request (unlike CGI requests).

**SSL accelerator** Hardware devices inserted just before the Web server that performs the computation-intensive step of decrypting encoded information to alleviate performance problems that sometimes accompany secure HTTP (HTTPS).

**think time** Automated test tools not only simulate the transactions that users make against the EnterpriseOne system. They also simulate the idle time in between each transaction where the user pauses to think. The duration of the think time is configurable.

**vertical scaling** Increasing the power of each physical machine (rather than adding more machines). Vertical scaling is typically used to allow more transactions per single user.

**virtual private network (VPN)** A technique of using encryption that allows a private network to be simulated over the public Internet. In effect, the wires of the public Internet are used, but thanks to encryption, no one on the public Internet can see the contents of the confidential information that crosses the wires.

Archived

# Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

## IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 114. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *IBM @server i5 and iSeries System Handbook*, GA19-5486
- ▶ *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014
- ▶ *WebSphere Edge Server: Working with Web Traffic Express and Network Dispatcher*, SG24-6172
- ▶ *IBM WebSphere V4.0 Advanced Edition Handbook*, SG24-6176
- ▶ *IBM WebSphere V5.1 Performance, Scalability, and High Availability: WebSphere Handbook Series*, SG24-6198
- ▶ *JD Edwards EnterpriseOne 8.9: High Availability, Storage, and the IBM @server pSeries*, SG24-6445
- ▶ *IBM WebSphere V5.0 Security Handbook: WebSphere Handbook Series*, SG24-6573
- ▶ *IBM WebSphere Portal V4.1 Handbook, Volume 1*, SG24-6883
- ▶ *IBM WebSphere V5.0 Edge of Network Patterns*, SG24-6896
- ▶ *WebSphere Application Server Enterprise V5 and Programming Model Extensions: WebSphere Handbook Series*, SG24-6932

## References

These publications are also relevant as further information sources:

- ▶ *Web Client Tuning Tips for iSeries*  
<http://www.ibm.com/support/docview.wss?uid=tss1wp100284>
- ▶ *IBM @server xSeries Performance and Tuning Tips for the J.D. Edwards Web Server*  
<http://www.ibm.com/support/docview.wss?uid=tss1wp100361>
- ▶ Concepts, Planning, and Installation for Edge Components on the IBM Information Center  
<http://www.ibm.com/software/webservers/appserv/doc/v51/ec/infocenter/index.html>
- ▶ Apache HTTP Server Version 1.3 documentation from Apache Software Foundation  
<http://www.apache.org>
- ▶ *Load Testing to Predict Web Performance* on the Mercury Interactive Corporation site  
<http://www.mercury.com/us/products/performance-center/loadrunner/papers.html>

- ▶ The following documents located in the Oracle PeopleSoft Customer/Partner Connection:  
<http://www.oracle.com/peoplesoft/integration.html>
  - *Audit Criteria for Certified Benchmarks* by JD Edwards Platform Technologies, with Harry Doby and Bill Calkins of Oracle
  - *Certifications Criteria for Published Benchmark Results* by JD Edwards Platform Technologies with Harry Doby and Bill Calkins of Oracle
  - *HTML Benchmark Kit: Test Methodology* by JD Edwards Platform Technologies, with Harry Doby and Bill Calkins of Oracle
  - *NetCentric Benchmarking* by JD Edwards Platform Technologies, with Harry Doby and Bill Calkins of Oracle
- ▶ Francis, Tim; Herness, Eric; High Jr., Rob; Knutson, Jim; Rochat, Kim; Vignola, Chris. *IBM WebSphere 5.0 Application Server*. Peer Information, Inc., December 2002. ISBN 1-86100-581-4.

## Online resources

These Web sites and URLs are also relevant as further information sources:

- ▶ IBM On Demand Business offering  
[http://www.ibm.com/services/ondemand/start\\_overview.html](http://www.ibm.com/services/ondemand/start_overview.html)
- ▶ IBM business continuity and recovery services  
<http://www.ibm.com/services/continuity/recover1.nsf/documents/home>
- ▶ OS/400 architecture  
<http://www.ibm.com/servers/enable/site/porting/iseres/overview/overview.html>

## How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

[ibm.com/redbooks](http://ibm.com/redbooks)

## Help from IBM

IBM Support and downloads

[ibm.com/support](http://ibm.com/support)

IBM Global Services

[ibm.com/services](http://ibm.com/services)

# Index

## A

- action items 105
- AIX 1, 8, 48, 53–54, 57–58, 60, 62, 76, 110
- allow thread allocation 54
  - beyond maximum 55
- application server or app server 29
- architecture
  - customization 29
  - logical 13
  - test 28

## B

- back-end software components 2
- BCS (Business Continuity Services) 19
- benchmark
  - iSeries 66
  - pSeries 48
  - xSeries 65
- best practices 83
- business case 40
  - for high availability 39
- Business Continuity Services (BCS) 19
- business objective, sample 92

## C

- caching proxy 3
- call object kernels 52
- channel 82
- components of Technology Foundation xiii
- configuration requirements 81
- connection backlog 58
- continuous availability 17
- continuously available 17
- cost
  - indirectly to the business 14
  - of availability 41
  - of high availability 40
- Credential Vault 85
- customization 29

## D

- database administrator 103
- database optimization 75
- database tuning 75
- DB2 Universal Database (UDB) 75
- defining a logical architecture 5
- defining a physical topology 5
- degraded failover 14
- degraded mode 14
- degree of failover 14
- demilitarized zone (DMZ) 20
- denial of service attack 41

- development test 80
- diagram
  - Highly Secure, Continuously Available logical architecture 27
  - Highly Secure, Highly Available logical architecture 26
  - Standard Security, Continuously Available logical architecture 25
  - Standard Security, Highly Available logical architecture 24
  - test architecture 28
- Directory Server, IBM 29
- disaster recovery 18, 39
- DMZ (demilitarized zone) 20
- dropped transactions 53, 66–67
- dummy session 64
- Dynacache 63
- dynamic caching 62
- dynamic content 2

## E

- Edge Server 3, 33
- emergency fix channel 82
- EnterpriseOne
  - back-end software components 2
  - front end 2
  - software suite 1
  - Web Server component 52

## F

- failover, degraded and non-degraded 14
- fault tolerance 14
  - requirements 39
- firewall 34
- four-step process 5
- front end 2
- full specification for physical architecture 5

## H

- hardware platform 1
- heap size 52
- high availability 16
- highly available 17, 34
- highly secure 19, 42
- Highly Secure, Continuously Available logical architecture 27
- Highly Secure, Highly Available logical architecture 26, 34
- horizontal scaling 29, 48, 86
- HTTP daemons 53
- HTTP-related Server layer 2

## I

- I/O timeout 64
- IBM Directory Server 29
- IBM HTTP Server (powered by Apache) 72
- IBM HTTP Server parameter 53
- infrastructure topology sample 79
- installation 70
- installation planning 13
- integration layer 81
- Internet access 36
- IP spraying 38
- iSeries 21, 23

## J

- JAS servlets 52
- JDENet layer 52
- justification 102

## K

- keepalive 56
- keepalive timeout 63
- kernel processes 74
- kernels 52

## L

- large project channel 82
- latency 32
- layer
  - Edge Server 2
  - HTTP server 2
  - Security Mechanisms 2
  - WebSEAL 2
  - WebSphere components 2
- LDAP 29
  - database 75
  - requirements 42
- licensing 3
- licensing agreement 3
- load balancer 3
- load balancing 38
- logical architecture
  - definition 5
  - fault tolerance 14
  - highly secure 20
  - highly secured 21
  - sample 89
  - security 14
  - selection 13
  - standard security 20
- Lotus collaboration administrator 104

## M

- maintenance 70, 76
- management tools 71
- max keepalive requests 64
- max thread size 54
- MaxClients 52–53

- method for Technology Foundation 7
- min and max spare servers 63

## N

- native.log file 55
- ndcontrol utility 71
- network and security administrator 103
- Network Dispatcher 3, 38, 71
- next numbers table 52, 66
- non-degraded failover 14
- non-degraded mode 15

## O

- operations monitoring tools 71
- OS/400 1, 8, 21, 66, 76, 110

## P

- parameters
  - disk access 47
  - memory 47
- performance 32, 36, 41
- Performance Viewer 74
- physical architecture specification 5
- physical topology definition 5
- planning 13
- Portal 74
- Portal administrator 104
- Portal Credential Vault 85
- Portal developer 103
- portlet 84
- post-installation 45
- pre-installation 5
- Process Choreographer 87
- process modeling 87
- producing a physical architecture 5
- production layer 83
- proposal submission 5
- pSeries 23
  - benchmark 48
- publicly accessible enterprises 85

## Q

- quality assurance environment 76
- quality of service 76

## R

- recommendation
  - long-term, sample 100
  - medium-term, sample 99
- recommended parameter values 52
- Redbooks Web site 114
  - Contact us xv
- redundancy 18, 39
- remote location 32
- Resource Analyzer 74
- response time 59
- reverse proxy elements 21

roles for installation and maintenance 70

## S

sample infrastructure topology 79

sample logical architecture 89

scripts 52

Secure Sockets Layer (SSL) 38

security 14, 41

    logical architecture 21

    standard 19

Security Mechanisms layer 2

serialized objects 52

Server Administration Workbench 74

server monitor 72

servlet caching 65

single level storage 21

skill set 70

    required 102

small project channel 82

software suite 1

SSL (Secure Sockets Layer) 38

SSL accelerators 38

standard security 42

Standard Security, Continuously Available logical architecture 25

Standard Security, Highly Available logical architecture 24

static content 62

static HTML 2

submitting the proposal 5

supported logical architectures 23

symbols 77

system administrator 104

system monitoring 76

## T

Technology Foundation

    components xiii

    enhancing with portlets 84

    layers 2

    maintenance 76

    methods and requirements 7

    Version 4 4, 23

    Version 5 4

test

    architecture 28, 49

    development 80

    scripts 66–67

    unit 80

Test Layer 82

thread allocation algorithm 55

thread inactivity 65

Tivoli Access Manager 21

Tivoli Performance Viewer 74

transaction inactivity timeout 65

transaction timeout 65

tuning

    database configuration 66

    HTTP Server 53

next number table 66

parameters 47, 53

WebSphere 48

## U

unit test 80

unplanned downtime 39

usersession timeout 59

## V

Version 4 Technology Foundation 3–4, 23

Version 5 Technology Foundation 3–4, 23

vertical scaling 29, 86

very highly secured 34

virtual address space 21

## W

Web Traffic Express 3

WebSEAL 21, 34, 71

WebSphere 73

WebSphere administrator 70, 103

WebSphere Business Integration Workflow 87

WebSphere components 2

WebSphere keepalives 56

WebSphere MQ Workflow 87

WebSphere Performance Package 3

WebSphere thread allocation 55

WebSphere tuning 59

## X

xSeries 23

Archived











# IBM *@*server and JD Edwards EnterpriseOne Technology Foundation

## Ensuring a High Quality of Service



**Redbooks**

**Configure Technology Foundation to be fast and maintainable**

**Protect your enterprise from expensive downtime**

**Secure your sensitive corporate and customer information**

Technology Foundation from Oracle is an integration of the IBM world-class middleware with JD Edwards EnterpriseOne application software. Technology Foundation provides a robust Web interface complete with collaboration and portal technology. This IBM Redbook covers the Technology Foundation components.

Part one discusses the steps to help you prepare for installing Technology Foundation. It helps you to determine your failover and security requirements prior to ordering hardware. Plus, it explains how to select a supported architecture for JD Edwards' EnterpriseOne Technology Foundation that:

- ▶ Protects your enterprise from expensive downtime
- ▶ Secures your sensitive corporate and customer information
- ▶ Is properly configured, fast, and maintainable

Part two examines issues that arise after you install Technology Foundation. It provides guidance to help you configure and tune Technology Foundation for high transaction volumes. This part also provides best practices to help you manage and maintain Technology Foundation.

This IBM Redbook is written for clients who want to improve the availability of their business. It also targets Oracle clients who are preparing to purchase EnterpriseOne Technology Foundation. It is also designed to help hardware vendors who work with the JD Edwards Technology Foundation clients.

### **INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION**

### **BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE**

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

**For more information:**  
[ibm.com/redbooks](http://ibm.com/redbooks)

SG24-6308-01

ISBN 0738493724