Sun Java System Portal Server 7 Deployment Planning Guide

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819–2582

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REPONDRE A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.

060210@13996

Contents

Preface 15

1 **Introduction to Portal Server Deployment Planning** 19
About Portal Server 19
Portal Server Features 20
Identity Management 20
Secure Remote Access 22
Search Engine 24
Personalizing Content 25
Aggregation and Integration 26
About Deployment Planning 27
Solution Life Cycle 27

2 **Business Analysis** 29
About Business Analysis 29
Defining Business Requirements 29
Understanding User Behaviors and Patterns 31

3 **Technical Requirements** 33
About Technical Requirements 33
Usage Analysis for Portal Server 34
Peak Numbers 34
Average Time Between Page Requests 35
Concurrent Users 35
Average Session Time 36
Search Engine Factors 36
Page Configuration 37
Portal Desktop Configuration 37

Hardware and Applications 38

Back-End Servers 38

Transaction Time 38

Workload Conditions 39

Usage Analysis for SRA 39

Scalability 40

Using SRA Prototype Numbers 40

Key Performance Factors 40

Portal Server Use Cases 42

Elements of Portal Server Use Cases 42

Example Use Case: Authenticate Portal User 43

Quality of Service Requirements 44

Performance 44

Availability 45

Scalability 45

Security 46

Designing for Localization 48

4 Logical Design 49

About Logical Architectures 49

Analysis of Logical Architecture 49

Portal Server Components 51

Secure Remote Access Components 52

SRA Gateway 53

Netlet 55

Netlet Proxy 57

NetFile 57

Rewriter 60

▼ Route for SRA Requests 60

Rewriter Proxy 61

Proxylet 61

Portal Server Nodes 62

Portal Server and Access Manager on Different Nodes 62

Portal Server System Communication Links 63

Example Portal Server Logical Architectures 66

A Typical Portal Server Installation 66

Portal Server Building Modules 67
Access Manager and Portal Server on Separate Nodes 75
Two Portal Servers One Access Manager 76
Two Portal Servers Two Access Managers 76
One Load Balancer Two Access Managers 77
Example SRA Logical Architectures 78
Basic SRA Configuration 79
Disable Netlet 80
Proxylet 81
Multiple Gateway Instances 82
Netlet and Rewriter Proxies 83
Netlet and Rewriter Proxies on Separate Nodes 85
Two Gateways and Netlet Proxy 86
Gateway with Accelerator 87
Netlet with 3rd Party Proxy 88
Reverse Proxy 89
Deployment Scenario 90

5 Deployment Design 91
About Deployment Design 91
Deployment Design Methodology 92
Estimating Processor Requirements 92
▼ Steps to Estimate Processor Requirements 92
Customize the Baseline Sizing Figures 93
Validate Baseline Sizing Figures 93
Refine Baseline Sizing Figures 94
Validate Your Final Figures 95
Identifying Performance Bottlenecks 95
Memory Consumption and Garbage Collection 95
Optimizing Resources 96
Hardware Accelerator 96
Sun Enterprise Midframe Line 96
Managing Risks 97

6 Implementation of a Deployment Design 99
About Implementing Deployment Designs 99

Documenting the Portal 100
Developing a Portal Prototype 100
▼ To Develop a Portal Prototype 100
Access Manager Cache and Sessions 102
Thread Usage 102
Portal Usage Information 102
Identity and Directory Structure Design 103
Creating a Custom Access Manager Service 103
Portal Desktop Design 104
Choosing and Implementing the Correct Aggregration Strategy 104
Working with Providers 104
Software Deployment 106
Software Packaging 106
Software Categories 106
Compatibility With Java Software 107
Client Support 107
Content and Design Implementation 108
Placement of Static Portal Content 108
Integrating Applications 108
Rolling Out a Production Deployment 110

7 Installed Product Layout 111
Directories Installed for Portal Server 111
Directories Installed for SRA 112
Configuration Files 112
Comparison of Solaris and Linux Path Names 113

8 Analysis Tools 115
Basic Tuning Techniques 115
mpstat 116
Output 116
iostat 117
Output 117
netstat 118
netstat -I hme0 10 118
Tuning Parameters for `/etc/system` 120

9 Portal Server and Application Servers 123

Application Server Support in Portal Server 123

Portal Server on an Application Server Cluster 124

▼ To Install Portal Server on an Application Server 124

Application Server Enterprise Edition 124

BEA WebLogic Server 125

IBM WebSphere Application Server 126

10 Troubleshooting Your Portal Deployment 127

Troubleshooting Portal Server 127

UNIX Processes 127

Log Files 128

Recovering the Search Database 128

▼ To Recover the Database 128

Working with the Display Profile 128

▼ To Extract the Display Profile 128

▼ To Reload the Display Profile 129

High CPU Utilization for Portal Server Instance 129

Configuring an HTTP Proxy 129

▼ To Configure Usage of an HTTP Proxy for a Portal Server Instance 129

Troubleshooting SRA 130

Debugging the Gateway 130

▼ To Use Debugging 130

SRA Log Files 131

11 Portal Deployment Worksheets 133

Portal Assessment Worksheets 133

Portal Design Task List 137

Index 143

Figures

FIGURE 1–1 Solution Life Cycle 28

FIGURE 4–1 Portal Server Communication Links 64

FIGURE 4–2 High-level Architecture for a Business-to-Employee Portal 67

FIGURE 4–3 SRA Deployment 67

FIGURE 4–4 Portal Server Building Module Architecture 68

FIGURE 4–5 Best Effort Scenario 70

FIGURE 4–6 No Single Point of Failure Example 71

FIGURE 4–7 Transparent Failover Example Scenario 73

FIGURE 4–8 Access Manager and Portal Server on Different Nodes 75

FIGURE 4–9 Two Portal Servers and One Access Manager 76

FIGURE 4–10 Two Portal Servers and Two Access Managers 77

FIGURE 4–11 Load Balancing two Access Managers 78

FIGURE 4–12 Basic SRA Configuration 80

FIGURE 4–13 Disable Netlet 81

FIGURE 4–14 Proxylet 82

FIGURE 4–15 Multiple Gateway Instances 83

FIGURE 4–16 Netlet and Rewriter Proxies 85

FIGURE 4–17 Proxies on Separate Nodes 86

FIGURE 4–18 Two Gateways and Netlet Proxy 87

FIGURE 4–19 SRA Gateway with External Accelerator 88

FIGURE 4–20 Netlet and Third-Party Proxy 89

FIGURE 4–21 Using a Reverse Proxy in Front of the Gateway 90

Tables

TABLE 1–1 Identity Management Features and Benefits 21

TABLE 1–2 SRA Features and Benefits 23

TABLE 1–3 Search Features and Benefits 24

TABLE 1–4 Personalization Features and Benefits 25

TABLE 1–5 Aggregation Features and Benefits 26

TABLE 3–1 Use Case: Authenticate Portal User 43

TABLE 4–1 Portal Server High Availability Scenarios 69

TABLE 7–1 Portal Server Directories 111

TABLE 7–2 Portal Server SRA Directories 112

TABLE 7–3 Comparison of Solaris and Linux Path Names 113

TABLE 8–1 Performance Analysis Tools 115

TABLE 8–2 /etc/system Options 120

TABLE 8–3 TCP/IP Options 121

TABLE 11–1 General Questions 134

TABLE 11–2 Organizational Questions 134

TABLE 11–3 Business Service-level Expectations Questions 135

TABLE 11–4 Content Management Questions 135

TABLE 11–5 User Management and Security Questions 135

TABLE 11–6 Business Intelligence Questions 136

TABLE 11–7 Architecture Questions 136

TABLE 11–8 Design Task List 137

Examples

Preface

The *Sun Java System Portal Server 7 Deployment Planning Guide* provides an introduction to planning and designing deployment solutions for Sun Java™ System Portal Server and Sun Java System Portal Server Secure Remote Access.

This guide assumes you have read the *Sun Java Enterprise System 2005Q4 Deployment Planning Guide* which presents basic concepts and principles of deployment planning and design, and discusses the solution life cycle, which encapsulates the phases and tasks of a deployment design project.

Who Should Use This Book

This guide is primarily intended for deployment architects and business planners responsible for the analysis and design of portal server deployments. This guide is also useful for system integrators and others responsible for the design and implementation of various aspects of an enterprise application.

Before You Read This Book

This guide assumes that you have read the*Sun Java Enterprise System 2005Q4 Deployment Planning Guide*

How This Book Is Organized

This guide is based on a solution life cycle which describes the various phases of deployment planning. The *Sun Java Enterprise System 2005Q4 Deployment Planning Guide*.

Portal Server Documentation Set

The Sun Java System Portal Server manuals are available as online files in Portable Document Format (PDF) and Hypertext Markup Language (HTML) formats. Both formats are readable by technologies for users with disabilities. The Portal Server documentation can be accessed here:

http://docs.sun.com/coll/1303.1

The following table lists the books included in the Portal Server 7 core documentation set.

Book Title	Description
Sun Java System Portal Server 7 Configuration Guide	Describes how to administer Portal Server 7 using the Portal Server administration console.
Sun Java System Portal Server 7 Installation Guide	Describes how to install Portal Server after Java Enterprise System has been installed.
Sun Java System Portal Server 7 Release Notes	Available after the product is released. Contains last-minute information, including a description of what is new in this current release, known problems and limitations, installation notes, and how to report issues with the software or the documentation.
Sun Java System Portal Server 7 Technical Overview	Provides an introduction to Portal Server concepts, architecture, and components.
Sun Java System Portal Server Developer Sample Guide	Provides information about how to create and deploy a Portal Server 7.0 developer sample desktop.
Sun Java System Portal Server Developer's Guide	Provides a high-level overview of the Sun Java System Portal Server software APIs.

Typographic Conventions

The following table describes the typographic changes that are used in this book.

TABLE P–1 Typographic Conventions

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`

TABLE P–1 Typographic Conventions *(Continued)*

Typeface	Meaning	Example

`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`**
		`Password:`

AaBbCc123	A placeholder to be replaced with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized (note that some emphasized items appear bold online)	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file.

Symbol Conventions

The following table explains symbols that might be used in this book.

TABLE P–2 Symbol Conventions

Symbol	Description	Example	Meaning
`[ ]`	Contains optional arguments and command options.	`ls [-l]`	The `-l` option is not required.
`{ | }`	Contains a set of choices for a required command option.	`-d {y|n}`	The `-d` option requires that you use either the `y` argument or the `n` argument.
`${ }`	Indicates a variable reference.	`${com.sun.javaRoot}`	References the value of the `com.sun.javaRoot` variable.
-	Joins simultaneous multiple keystrokes.	Control-A	Press the Control key while you press the A key.
+	Joins consecutive multiple keystrokes.	Ctrl+A+N	Press the Control key, release it, and then press the subsequent keys.
→	Indicates menu item selection in a graphical user interface.	File → New → Templates	From the File menu, choose New. From the New submenu, choose Templates.

Accessing Sun Resources Online

The docs.sun.com^{SM} web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. Books are available as online files in PDF and HTML formats. Both formats are readable by assistive technologies for users with disabilities.

To access the following Sun resources, go to `http://www.sun.com`:

- Downloads of Sun products
- Services and solutions
- Support (including patches and updates)
- Training
- Research
- Communities (for example, Sun Developer Network)

Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

Note – Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. To share your comments, go to `http://docs.sun.com` and click Send Comments. In the online form, provide the full document title and part number. The part number is a 7-digit or 9-digit number that can be found on the book's title page or in the document's URL. For example, the part number of this book is 819-2582.

◆ ◆ ◆ CHAPTER 1

Introduction to Portal Server Deployment Planning

This chapter provides a brief overview of Sun Java™ System Portal Server and Sun Java System Portal Server Secure Remote Access. This chapter introduces the solution life cycle, which outlines the various steps for planning and designing enterprise software systems. See the *Sun Java Enterprise System 2005Q4 Deployment Planning Guide*for information on basic concepts and principles of deployment planning and design.

This chapter contains the following sections:

- "About Portal Server" on page 19
- "Portal Server Features" on page 20
- "About Deployment Planning" on page 27
- "Solution Life Cycle" on page 27

About Portal Server

Portal Server is a component of the Sun Java Enterprise System technology. Sun Java Enterprise System technology supports a wide range of enterprise computing needs, such as creating a secure intranet portal to provide the employees of an enterprise with secure access to email and in-house business applications.

The Portal Server product is an identity-enabled portal server solution. It provides all the user, policy, and identity management to enforce security, web application single sign-on (SSO), and access capabilities to end user communities. In addition, Portal Server combines portal services, such as personalization, aggregation, security, integration, and search. Unique capabilities that enable secure remote access to internal resources and applications round out a complete portal platform for deploying business-to-employee, business-to-business, and business-to-consumer portals. Sun Java System Portal Server Secure Remote Access (SRA) provides secure remote access capabilities to access web- and non-web enabled resources.

Each enterprise assesses its own needs and plans its own deployment of Java Enterprise System technology. The optimal deployment for each enterprise depends on the type of applications that Java Enterprise System technology supports, the number of users, the kind of hardware that is available, and other considerations of this type.

Portal Server is able to work with previously installed software components. In this case, Portal Server uses the installed software when the software is an appropriate version.

Portal Server runs in open mode or secure mode. In secure mode, it uses Secure Remote Access (SRA). The main difference between an open portal and a secure portal is that the services presented by the open portal typically reside within the demilitarized zone (DMZ) and not within the secured intranet. SRA offers browser-based secure access to portal content and services from any remote browser enabled with Java technology. Integration with Portal Server software ensures that users receive secure encrypted access to the content and services that users have permission to access.

SRA is targeted toward enterprises deploying highly secure remote access portals. These portals emphasize security, protection, and privacy of intranet resources. The SRA services–Access List, the Gateway, NetFile, Netlet, and Proxylet– enable users to securely access intranet resources through the Internet without exposing these resources to the Internet.

Portal Server Features

This section reviews specific technology features with the goal of determining which technologies are most important for your organization. Review these features while keeping in mind your organization's short, mid, and long term plans.

Use the following sections and tables to assess the benefits of the listed features and determine their relative priority for your organization. This information will assist you in developing a deployment plan in a timely and cost effective manner.

Note – In all likelihood, your Java ™ Enterprise System sales representative has previously discussed these topics with you. Thus, this section serves as a review of that process.

Identity Management

Portal Server uses identity management to control many users spanning a variety of different roles across the organization and sometimes outside the organization while accessing content, applications and services. The challenges include: Who is using an application? In what capacity do users serve the organization or company? What do users need to do, and what should users be able to access? How can others help with the administrative work?

Table 1–1 shows the identity management features and their benefits.

TABLE 1–1 Identity Management Features and Benefits

Feature	Description	Benefit
Directory service	Portal Server uses Access Manager and Directory Server	Portal Server uses an LDAP directory for storing user profiles, roles, and identity information for the purpose of authentication, single sign-on (SSO), delegated administration, and personalization Portal Server uses an open schema that can reside in a centralized user directory, thereby leveraging an enterprise or service provider's investment in the Sun Java System Access Manager and Sun Java System Directory Server products.
User, policy, and provisioning management	Access Manager enables you to manage many users spanning a variety of different roles across the organization and sometimes outside the organization while accessing content, applications, and services.	Provides a centralized identity management solution for storing and managing identity information, which is integrated with a policy solution to enforce access rights, greatly simplifying these challenges. Extends a common identity to handle new applications, enables applications to share administrative work, and simplifies tasks normally associated with building these services. Consolidates management of users and applications. Personalizes content and service delivery. Simplifies and streamlines information and service access. Reduces costs associated with managing access and delivery. Provides secure policy-based access to applications. Ensures secure access as portal deployments expand beyond employee LAN access.

TABLE 1–1 Identity Management Features and Benefits *(Continued)*

Feature	Description	Benefit
Single sign-on (SSO)	Access Manager integrates user authentication and single sign-on through an SSO API. Once the user is authenticated, the SSO API takes over. Each time the authenticated user tries to access a protected page, the SSO API determines if the user has the permissions required based on their authentication credentials. If the user is valid, access to the page is given without additional authentication. If not, the user is prompted to authenticate again.	Enhances user productivity by providing a consistent, centralized mechanism to manage authentication and single sign-on, while enabling employees, partners and customers access to content, applications, and services.
Delegated administration	The Portal Server administration console provides role-based delegated administration capabilities to different kinds of administrators to manage organizations, users, policy, roles, channels, and Portal Desktop providers based on the given permissions.	Enables IT to delegate portal administrative duties to free up valuable IT resources and administration.
Security	Provides single sign-on for aggregated applications to the portal.	Security is an important functionality in portals. Security can address many different needs within the portal, including authentication into the portal, encryption of the communications between the portal and the end user, and authorization of the content and applications to only users that are allowed access.

Secure Remote Access

Table 1–2 shows the Sun Java System Portal Server Secure Remote Access (SRA) features and their benefits.

TABLE 1–2 SRA Features and Benefits

Feature	Description	Benefit
Integrated security	Extranet or Virtual Private Network capabilities “on demand” while providing user, policy, and authentication services. The Gateway component provides the interface and security barrier between remote user sessions originating from the Internet, and your corporate intranet.	Extends an enterprise’s content, applications, files, and services located behind firewalls to authorized suppliers, business partners, and employees. To prevent denial of service attacks, you can use both internal and external DMZ-based Gateways.
SRA core	Users achieve remote access through four components: ■ Gateway ■ NetFile ■ Netlet ■ Proxylet	This component has four parts: ■ Gateway—Controls communication between the Portal Server and the various Gateway instances. ■ NetFile—Enables remote access and operation of file systems and directories. ■ Netlet—Ensures secure communication between the Netlet applet on the client browser, the Gateway, and the application servers. ■ Proxylet—Proxylet sets itself up as a proxy server running on the client’s machine, and modifies the proxy settings of the browser to point to itself (also referred to as the local proxy server). The local proxy server (Proxylet) then proxies all the intranet traffic through the gateway.
Universal access	Enables web browser based universal access with no client software installation or maintenance necessary.	Simplifies the IT administration and maintenance overhead while dramatically reducing the time and cost of deployment.
Netlet Proxy	Provides an optional component that extends the secure tunnel from the client, through the Gateway to the Netlet Proxy that resides in the intranet.	Restricts the number of open ports in a firewall between the demilitarized zone (DMZ) and the intranet.

TABLE 1–2 SRA Features and Benefits *(Continued)*

Feature	Description	Benefit
Rewriter Proxy	Redirects HTTP requests to the Rewriter Proxy instead of directly to the destination host. The Rewriter Proxy in turn sends the request to the destination server.	Using the Rewriter Proxy enables secure HTTP traffic between the Gateway and intranet computers and offers two advantages: ■ If a firewall exists between the Gateway and server, the firewall needs to open only two ports: one between the Gateway and the Rewriter Proxy, and another between the Gateway and the Portal Server. ■ HTTP traffic is now secure between the Gateway and the intranet even if the destination server only supports HTTP protocol (no HTTPS).

Search Engine

The Search Engine service is used in the following channels:

- Subscription channel to summarize the number of hits (relevant information) that match each profile entry defined by the user for categorized documents and discussions.
- Discussion channel to individually search contents and rate the importance for comments.

Table 1–3 lists the search features and their benefits.

TABLE 1–3 Search Features and Benefits

Feature	Description	Benefit
Search Engine	Enables the retrieval of documents based on criteria specified by the end user.	Saves users time by providing access to content.
Categorization	Organizes documents into a hierarchy. This categorization is often referred to as taxonomy.	Provides a different view of documents that enables browsing and retrieval.
Robot	The Search Engine robot is an agent that crawls and indexes information across your intranet or the Internet.	Automatically searches and extracts links to resources, describes those resources, and puts the descriptions in the Search database (also called generation or indexing).

TABLE 1–3 Search Features and Benefits (Continued)

Feature	Description	Benefit
Discussions	A forum for multiple threaded discussions.	Contents are individually searchable and importance rating are given for of all comments
Subscriptions	Enables the user to track new or changed material in different areas of interest.	Discussions, search categories, and free-form searches (saved searches) can be tracked.

Personalizing Content

Personalization is the ability to deliver content based on selective criteria and offer services to a user.

Table 1–4 shows the personalization features and their benefits.

TABLE 1–4 Personalization Features and Benefits

Feature	Description	Benefit
Deliver content based on user's role	Portal Server includes the ability to automatically choose which applications users are able to access or to use, based on their role within the organization.	Increases employee productivity, improves customer relationships, and streamlines business relationships by providing quick and personalized access to content and services.
Enable users to customize content	Portal Server enables end users to choose what content they are interested in seeing. For example, users of a personal finance portal choose the stock quotes they would like to see when viewing their financial portfolio.	The information available in a portal is personalized for each individual. In addition, users can then customize this information further to their individual tastes. A portal puts control of the web experience in the hands of the people using the web, not the web site builders.
Aggregate and personalize content for multiple users	Portal Server enables an enterprise or service provider to aggregate and deliver personalized content to multiple communities of users simultaneously.	This enables a company to deploy multiple portals to multiple audiences from one product and manage them from a central management console. Also, new content and services can be added and delivered on demand without the need to restart Portal Server. All of this saves time and money, and ensures consistency in an IT organization.

Aggregation and Integration

One of the most important aspects of a portal is its ability to aggregate and integrate information, such as applications, services, and content. This functionality includes the ability to embed non-persistent information, such as stock quotes, through the portal, and to run applications within, or deliver them through, a portal.

Table 1–5 shows the aggregation and integration features and their benefits.

TABLE 1–5 Aggregation Features and Benefits

Feature	Description	Benefit
Aggregated information	The Portal Desktop provides the primary end-user interface for Portal Server and a mechanism for extensible content aggregation through the Provider Application Programming Interface (PAPI). The Portal Desktop includes a variety of providers that enable container hierarchy and the basic building blocks for building some types of channels.	Users no longer have to search for the information. Instead, the information finds them.
Consistent set of tools	Users can use the provided set of tools such as web-based email and calendaring software that follows them through their entire time at the company.	Users do not have to use one tool for one project, another tool for another location. Also, because these tools all work within the portal framework, the tools have a consistent look and feel and work similarly, reducing training time.
Collaboration	Portal Server provides control and access to data as a company-wide resource.	In many companies, data is seen as being owned by individual departments, instead of as a company-wide resource. The portal can act as a catalyst for breaking down these silos and making the data available in a controlled way to the people who need to use it. This broader, more immediate access can improve collaboration.
Integration	Portal Server enables you to use the Portal Desktop as the sole place for users to gain access to or launch applications and access data.	Integration with existing email, calendar, legacy, or web applications enables the portal to serve as a unified access point, enabling users—be that employees, partners, or customers—to access the information users need quickly and easily.

About Deployment Planning

Deployment planning is a critical step in the successful implementation of a Java Enterprise System solution. Each enterprise has its own set of goals, requirements, and priorities to consider. Successful planning begins with analyzing the goals of an enterprise and determining the business requirements to meet those goals. The business requirements must then be converted into technical requirements that can be used as a basis for designing and implementing a system that can meet the goals of the enterprise.

For information see the *Sun Java Enterprise System 2005Q4 Deployment Planning Guide*.

Solution Life Cycle

The solution life cycle shown in the following figure depicts the steps in the planning, design, and implementation of an enterprise software solution based on Java Enterprise System. The life cycle is a useful tool for keeping a deployment project on track.

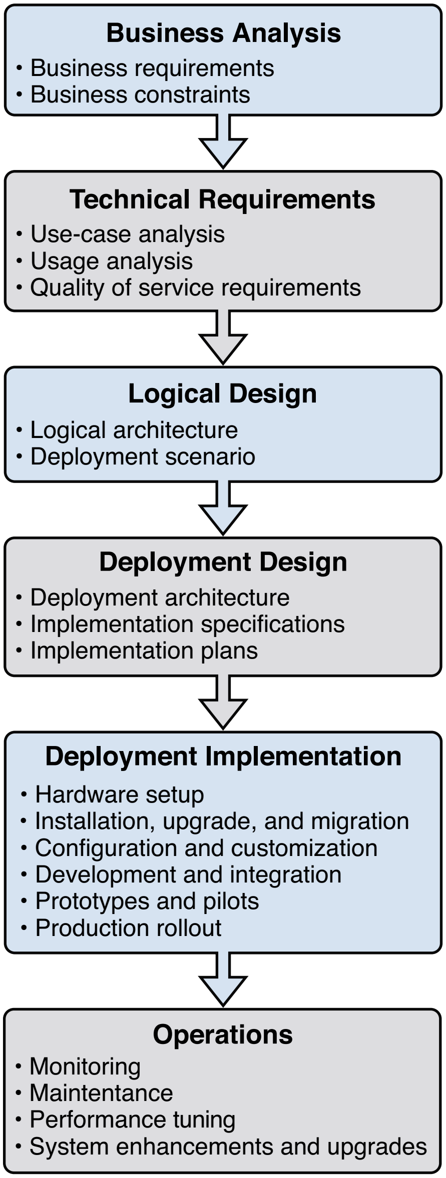

FIGURE 1–1 Solution Life Cycle

◆ ◆ ◆ CHAPTER 2

Business Analysis

During the business analysis phase of the solution life cycle you define business goals by analyzing a business problem and identifying the business requirements and business constraints to meet that goal.

This chapter contains the following sections:

- "About Business Analysis" on page 29
- "Defining Business Requirements" on page 29
- "Understanding User Behaviors and Patterns" on page 31

About Business Analysis

Business analysis starts with stating the business goals. You then analyze the business problems you must solve and identify the business requirements that must be met to achieve the business goals. Consider also any business constraints that limit your ability to achieve the goals. The analysis of business requirements and constraints results in a set of business requirements documents.

You use the resulting set of business requirements documents as a basis for deriving technical requirements in the technical requirements phase. Throughout the solution life cycle, you measure the success of your deployment planning and ultimately the success of your solution according to the analysis performed in the business analysis phase.

Defining Business Requirements

The business requirements of your portal affect deployment decisions. Understand your requirements to make correct assumptions that affect the accuracy of your deployment estimates.

The reasons you are offering your portal have a direct affect on how you implement your portal. You must define target population, performance standards, and other factors related to your goals.

Use these questions to help you identify the goals of your portal:

- What is your portal's biggest priority?
- What applications will the portal deliver?
- What is your target population?
- What performance standard is necessary?
- What transaction volume do you expect? What transaction volume do you expect during peak use?
- What response time is acceptable during peak use?
- What is the necessary level of concurrency? Concurrency is the number of users who can be connected at any given time?
- Should access to the portal be through the intranet or the Internet?
- Will your portal be deployed in one phase, or many phases?
- What are the business requirements of this portal? For example, do you want to enhance customer service? Increase employee productivity? Reduce the cost of doing business?
- What kind of portal do you need? For example, business-to-business, business-to-consumer, business-to-enterprise, or a hybrid?
- Who is your target audience?
- What services or functions will the portal deliver to users?
- How will the target audience benefit from the portal?
- What are the priorities for the portal?

(Optional) Use these questions to help identify your business objectives if you are deploying a secure portal:

- Do you need to increase employee productivity (by making your intranet applications and servers accessible over the Internet)?
- Do you need to provide secure access to your portal?
- Do you need to reduce the cost of ownership of an existing Virtual Private Network (VPN) solution?
- Do you want employees to access intranet applications such as Citrix and pcAnywhere from the Internet?
- Do you want your employees to explore intranet servers or machines from the Internet?
- Who is your target audience (all portal users, employees, or customers)?

Understanding User Behaviors and Patterns

Study the people who will use your portal. Factors such as when users will use the portal and how users have used predecessor systems are keys to identifying your requirements. If your organization's experience cannot provide these patterns, you can study the experience of other organizations and estimate them.

Use these questions to help you understand users:

- How many end users will you have? What is the size of your target audience?
- Will users login to the portal at the same time each day? Will they use the portal at work or somewhere else?
- Are users in the same time zone or in different time zones?
- How long do you expect the typical user to be connected, or have a valid portal session open? What use statistics do you have for existing applications? Do you have web traffic analysis figures for an existing portal?
- How many visitor sessions, or number of single-visitor visits, are likely within a predefined period of time?
- Is portal use likely to increase over time? Or stay stable?
- How fast will your user base grow?
- How have your users used an application that the portal will deliver to them?
- What portal channels do you expect users to use regularly?
- What expectations about your portal content do your users have? How have users used predecessor web-based information or other resources that your portal will offer?

CHAPTER 3

Technical Requirements

During the technical requirements phase of the solution life cycle you perform a usage analysis, identify use cases, and determine quality of service requirements for the proposed deployment solution.

This chapter contains the following sections:

- "About Technical Requirements" on page 33
- "Usage Analysis for Portal Server" on page 34
- "Usage Analysis for SRA" on page 39
- "Portal Server Use Cases" on page 42
- "Quality of Service Requirements" on page 44
- "Designing for Localization" on page 48

About Technical Requirements

Technical requirements analysis begins with the business requirements documents created during the business analysis phase of the solution life cycle. Using the business analysis as a basis, you do the following:

- Perform a usage analysis to aid in determining expected load conditions.
- Create use cases that model typical user interaction with the system.
- Create a set of quality of service requirements (QoS) that define how a deployed solution must perform in areas such as response time, availability, security, and others.

The quality of service requirements are derived from the usage analysis and the use cases, keeping in mind business requirements and constraints previously identified. The quality of service requirements are later paired with logical architectures in the logical design phase to form a deployment scenario. The deployment scenario is the main input to the deployment design phase of the solution life cycle.

As with business analysis, no simple formula for technical requirements analysis exists that generates the usage analysis, use cases, and system requirements. Technical requirements analysis requires an understanding of the business domain, business objectives, and the underlying system technology.

Usage Analysis for Portal Server

You need to establish baseline sizing figures that can be used in the logical and architecture and deployment design. Your technical representative can provide you with an automated sizing tool to calculate the estimated number of CPUs your Portal Server deployment requires.

Note – Sizing requirements for a secure portal deployment using Sun Java™ System Secure Remote Access (SRA) software are covered in "Usage Analysis for SRA" on page 39.

You need to gather the following metrics for input to the sizing tool:

- "Peak Numbers" on page 34
- "Average Time Between Page Requests" on page 35
- "Concurrent Users" on page 35
- "Average Session Time" on page 36
- "Search Engine Factors" on page 36
- "Page Configuration" on page 37

Other performance metrics that affect the number of CPUs a Portal Server deployment requires, but are not used by the sizing tool, are:

- Portal Desktop Configuration
- Hardware and Applications
- Back-end Servers
- Transaction Time
- Workload Conditions

A discussion of the these performance factors follows.

Peak Numbers

Maximum number of concurrent sessions defines how many connected users a Portal Server deployment can handle.

To calculate the maximum number of concurrent sessions, use this formula:

```
maximum number of concurrent sessions =
expected percent of users online * user base
```

To identify the size of the user base or pool of potential users for an enterprise portal, here are some suggestions:

- Identify only users who are active. Do not include users who are, for example, away on vacation, or on leave.
- Use a finite figure for user base. For an anonymous portal, estimate this number conservatively.
- Study access logs.
- Identify the geographic locations of your user base.
- Remember what your business plan states regarding who your users are.

Average Time Between Page Requests

Average time between page requests is how often, on average, a user requests a page from the Portal Server. Pages could be the initial login page to the portal, or a web site or web pages accessed through the Portal Desktop. A page view is a single call for a single page of information no matter how many items are contained on the page.

Though web server logs record page requests, using the log to calculate the average time between requests on a user basis is not feasible. To calculate the average time between page requests, you would probably need a commercially available statistics tool, such as the WebLoad performance testing tool. You can then use this figure to determine the number of concurrent users.

Note – Page requests more accurately measure web server traffic than “hits.” Every time any file is requested from the web server counts as a hit. A single page call can record many hits, as every item on the page is registered. For example, a page containing 10 graphic files records 11 “hits”—one for the HTML page itself and one for each of the 10 graphic files. For this reason, page requests gives a more accurate determination of web server traffic.

Concurrent Users

A *concurrent user* is one connected to a running web browser process and submitting requests to or receiving results of requests from Portal Server. The maximum number of concurrent users is the highest possible number of concurrent users within a predefined period of time. Calculate the maximum number of concurrent users after you calculate the maximum number of concurrent sessions. To calculate the maximum number of concurrent users, use this formula:

```
concurrent users = number of concurrent sessions / average time between hits
```

For example, consider an intranet Portal Server example of 50,000 users. The number of connected sessions under its peak loads is estimated to be 80% of its registered user base. On average, a user accesses the Portal Desktop once every 10 minutes.

The calculation for this example is:

```
40000 / 10 = 4000
```

The maximum number of concurrent users during the peak hours for this Portal Server site should be 4,000.

Average Session Time

Average session time is the time between user login and logout averaged over a number of users. The length of the session time is inversely proportional to the number of logins occurring (that is, the longer the session duration, the fewer logins per second are generated against Portal Server for the same concurrent users base). Session time is the time between user login and user logout.

How the user uses Portal Server often affects average session time. For example, a user session involving interactive applications typically has a longer session time than a user session involving information only.

Search Engine Factors

If your portal site will offer a Search channel, you need to include sizing factors for the Search Engine in your sizing calculations. Search Engine sizing requirements depend on the following factors:

- The size of index partitions on the active list of the index directory

  Partition size is directly proportional to the size and number of indexed and searchable terms.
- Average disk space requirement of a resource description (RD)

  To calculate this, use this formula:

  ```
  average disk space requirement =
  database size / number of RDs in database
  ```

  The average size adjusts for variations in sizes of RDs. A collection of long, complex RDs with many indexed terms and a list of short RDs with a few indexed terms require different search times, even if the complex RDs have the same number of RDs.

  RDs are stored in a hierarchical database format, where the intrinsic size of the database must be accounted for, even when no RD is stored.
- The number of concurrent users who perform search-related activities

  To calculate this, use this formula:

  ```
  number of concurrent users / average time between search hits
  ```

  Use the number of concurrent users value calculated in “Concurrent Users” on page 35.
- The type of search operators used

  Types of search functions include basic, combining, proximity, passage and field operator, and wildcard scans. Each function uses different search algorithms and data structures. Because differences in search algorithms and data structures increase as the number of search and indexed terms increase, the type of search function affects times for search result return trips.

Page Configuration

If you are using an authenticated portal, you must specify both Login Type and Desktop Type in the page configuration section of the automated sizing tool.

- **Login Type**. Describes the type of portal page (content configuration and delivery method) that end users initially see after submitting user name and password. This process s typically taxing on the system because the process involves checking credentials, initializing the session, and delivering initial content.

  The Measured CPU Performance characteristic associated with the Login Type is the *Initial Desktop Display* variable.
- **Desktop Type**. Describes the type of portal pages (content configuration and delivery method) that end users see after the initial portal page. These pages are displayed with each subsequent interaction with the portal, or on Desktop refresh. Because the session has already been established and cached content can be exploited, less system resources are typically required and the pages are delivered more rapidly.

  The Measured CPU Performance characteristic associated with the Desktop Type is the *Desktop Reload* variable.

For both Login Type and Desktop Type, select the appropriate content configuration:

- Light-JSP. Describes a configuration of two tabs with five channels each.
- Regular-JSP. Describes a configuration of two tabs with seven channels each.
- Heavy-JSP. Describes a configuration of three tabs with seventeen channels each.

Tip – You can now give the above figures to your technical representative and ask that the sizing tool be run to identify your estimated number of CPUs.

Portal Desktop Configuration

Portal Desktop configuration explicitly determines the amount of data held in memory on a per-session basis.

The more channels on the Portal Desktop, the bigger data session size, and the lesser the throughput of Portal Server.

Another factor is how much interactivity the Portal Desktop offers. For example, channel clicks can generate load on Portal Server or on some other external server. If channel selections generate load on Portal Server, a higher user activity profile and higher CPU overhead occur on the node that hosts the Portal Desktop than on a node that hosts some other external server.

Hardware and Applications

CPU speed and size of the virtual machine for the Java platform (Java Virtual Machine or JVM™ software) memory heap affect Portal Server performance.

The faster the CPU speed, the higher the throughput. The JVM memory heap size, along with the heap generations tuning parameters, can also affect Portal Server performance.

Back-End Servers

Portal Server aggregates content from external sources. If external content providers cannot sustain the necessary bandwidth for Portal Server to operate at full speed, Portal Desktop rendering and throughput request times will not be optimum. The Portal Desktop waits until all channels are completed (or timed out) before it returns the request response to the browser.

Plan your back-end infrastructure carefully when you use channels that:

- Scrape their content from external sources
- Access corporate databases, which typically have slow response times
- Provide email content
- Provide calendar content

Transaction Time

Transaction time, which is the delay taken for an HTTP or HTTPS operation to complete, aggregates send time, processing time, and response time figures.

You must plan for factors that can affect transaction time. These include:

- Network speed and latency.

  You need to especially examine latency over a Wide Area Network (WAN). Latency can significantly increase retrieval times for large amounts of data.
- The complexity of the Portal Desktop.
- The browser's connection speed.

  For example, a response time delay is longer with a connection speed of 33.6 kilobytes per second than with a LAN connection speed. However, processing time should remain constant. Transaction time through a dial-up connection should be faster than transaction time displayed by a load generation tool because it performs data compression.

When you calculate transaction time, size your Portal Server so that processing time under regular or peak load conditions does not exceed your performance requirement threshold and so that you can sustain processing time over time.

Workload Conditions

Workload conditions are the most predominantly used system and JVM software resources on a system. These conditions largely depend on user behavior and the type of portal you deploy.

The most commonly encountered workload conditions on Portal Server software affect:

- System performance

  Portal Server performance is impacted when a large number of concurrent requests are handled (such as a high activity profile). For example, during peak hours in a business-to-enterprise portal, a significant number of company employees connect to the portal at the same time. Such a scenario creates a CPU-intensive workload. In addition, the ratio of concurrent users to connected users is high.

- System capacity

  Portal Server capacity begins to be impacted when large numbers of users log in. As more users login, users use more of the available memory, and subsequently, less memory is available to process requests made to the server. For example, in a business-to-consumer web portal, a large number of logged-in users are redirected to external web sites once the initial Portal Desktop display is loaded. However, as more users continue to login, users create the need for more memory, even though the ratio of users submitting requests to Portal Server and the users merely logged-in is low.

  Depending on the user's behavior at certain times of the day, week, or month, Portal Server can switch between CPU-intensive and memory-intensive workloads. The portal site administrator must determine the most important workload conditions to size and tune the site to meet the enterprise's business goals.

Usage Analysis for SRA

Use this section only if your organization is implementing a secure portal by installing SRA. As you did for portal, for SRA, you must first establish your Gateway instances baseline sizing estimate (A single machine can have one Gateway installation but multiple instances. SRA enables you to install multiple Gateways, each running multiple instances.) Your design decisions help you make accurate estimates regarding SRA user sessions and concurrency.

You must first establish your Gateway instances baseline sizing estimate. This baseline figure represents what you must have to satisfy your Gateway user sessions and concurrency needs.

Establishing an appropriate sizing estimate for your SRA deployment is an iterative process. You might wish to change the inputs to generate a range of sizing results. Test these results against your original requirements. You can avoid most performance problems by formulating your requirements correctly and setting realistic expectations of SRA performance.

Note – Properly sizing the Gateway is difficult, and using the Gateway sizing tool is only the beginning. Gateway performance depends more on throughput then on the number of users, active users, or user sessions. Any sizing information for the Gateway has to be based on a set of assumptions.

Scalability

You can choose between one, two, and four CPUs per Gateway instance. The number of CPUs bound to a Gateway instance determines the number of Gateway instances required for the deployment.

Using SRA Prototype Numbers

If you have numbers from a prototype of the portal with SRA, you can use these numbers in the Gateway sizing tool to arrive at more accurate results. You would fill in the following:

- Measured CPU Performance. The values used to help calculate the number of Gateway instances include:
  - Initial Portal Desktop Display, hits per second per CPU
  - Portal Desktop Reloads, hits per second per CPU
- Netlet Applications Block Size. This value specifies the Netlet application byte size. The Netlet dynamically determines the block size based on the application that is used. Block size determined by Netlet for a Telnet is based on the amount of data transferred.

Note – You do not need to specify the Page Configuration and Scalability options if you are using trial deployment numbers.

Key Performance Factors

Key performance factors are metrics that your technical representative uses as input to an automated sizing tool. The sizing tool calculates the estimated number of Gateway instances your SRA deployment requires.

Identifying these key performance factors and giving them to your technical representative is the first step in formulating your baseline sizing figure.

These are the key performance factors:

- "Session Characteristics" on page 41

- "Netlet Usage Characteristics" on page 41

Note – After you calculate these key performance factors, give the figures to your technical representative. Ask that the Gateway sizing tool be run to identify the estimated number of Gateway instances.

Session Characteristics

The session characteristics of the Gateway include:

- Total number of SRA (Gateway) users

  This represents the size of your user base or pool of potential users for the secure portal..
- Expected percentage of total users using the Gateway (at maximum load)

  Apply a percentage to your total number of users to determine this figure.
- Average time between page hits

  This is how often on average a user requests a page from the portal server.
- Session average time

  This determines how many logins per second that the Gateway must sustain for a given number of concurrent users.

Netlet Usage Characteristics

Consider the following Netlet characteristics of the Gateway, which can have a impact on calculating the number of Gateway instances:

- Netlet is enabled in the Portal Server administration console.

  If Netlet is enabled, the Gateway needs to determine whether the incoming traffic is Netlet traffic or Portal Server traffic. Disabling Netlet reduces this overhead since the Gateway assumes that all incoming traffic is either HTTP or HTTPS traffic. Disable Netlet only if you are sure you do not want to use any remote applications with Portal Server.
- Expected percentage of total users using Netlet

  Apply a percentage to your total number of users to determine this figure.
- Expected throughput

  Determine the expected throughput of your Gateway, expressed in kilobits per second (Kbps).
- Netlet Cipher (encryption) being used

  Choices include Native VM and Java software plugin ciphers.

Portal Server Use Cases

Use cases are written scenarios used to test and present the system's capabilities and form an important part of your high-level design. Though you implement use case scenarios toward the end of the project, formulate them early in the project once you have established your requirements.

When available, use cases can provide valuable insight into how the system is to be tested. Use cases are beneficial in identifying how you need to design the user interface from a navigational perspective. When designing use cases, compare them to your requirements to get a thorough view of their completeness and how you are to interpret the test results.

Use cases provide a method for organizing your requirements. Instead of a bulleted list of requirements, you organize them in a way that tells a story of how someone can use the system. This provides for greater completeness and consistency, and also gives you a better understanding of the importance of a requirement from a user perspective.

Use cases help to identify and clarify the functional requirements of the portal. Use cases capture all the different ways a portal would be used, including the set of interactions between the user and the portal as well as the services, tasks, and functions the portal is required to perform.

A use case defines a goal-oriented set of interactions between external actors and the portal system. (Actors are parties outside the system that interact with the system, and can be a class of users, roles users can play, or other systems.)

Use case steps are written in an easy-to-understand structured narrative using the vocabulary of the domain.

Use case scenarios are an instance of a use case, representing a single path through the use case. Thus, there may be a scenario for the main flow through the use case and other scenarios for each possible variation of flow through the use case (for example, representing each option).

Elements of Portal Server Use Cases

When developing use cases for your portal, keep the following elements in mind:

- **Priority**. Describes the priority, or ranking of the use case. For example, this could range from high to medium to low.
- **Context of use**. Describes the setting or environment in which the use case occurs.
- **Scope**. Describes the conditions and limititations of the use case.
- **Primary user**. Describes what kind of user this applies to, for example, an end user or an administrator.
- **Special requirements**. Describes any other conditions that apply.
- **Stakeholders**. Describes the people who have a vested interest in how a product decision is made or carried out.
- **Precondition**. Describes the prerequisites that must be met for the use case to occur.

- **Minimal guarantees**. Describes the minimum that must occur if the use case is not successfully completed.
- **Success guarantees**. Describes what happens if the use case is successfully completed.
- **Trigger**. Describes the particular item in the system that causes the event to occur.
- **Description**. Provides a step-by-step account of the use case, from start to finish.

Example Use Case: Authenticate Portal User

Table 3–1 describes a use case for a portal user to authenticate with the portal.

TABLE 3–1 Use Case: Authenticate Portal User

Item	Description
Priority	Must have.
Context of Use	Only authenticated users are allowed to gain access to the portal resources. This access restriction applies to all portal resources, including content and services. This portal relies on the user IDs maintained in the corporate LDAP directory.
Scope	The portal users identify themselves only once for a complete online session. In the case that an idle time-out occurs, the users must reidentify themselves. If the portal user identification fails more often than a specified amount of allowed retries, access to the intranet should be revoked or limited (deactivated) until a system administrator reactivates the account. In this case, the portal user should be advised to contact the authorized person. The identified portal users are able to access only the data and information that they are authorized for.
Primary User	Portal end user.
Special Requirements	None.
Stakeholders	Portal end user.
Preconditions	The portal user is an authorized user. Standard corporate LDAP user ID. Must be provided to each employee. Authorized LDAP entry. Every employee has access to the corporate intranet. No guest account.
Minimal Guarantees	Friendly customer-centric message. Status—with error message indicating whom to call.
Success Guarantees	Presented with Portal Desktop home page. Authentication. Entitlement. Personal information.
Trigger	When any portal page is accessed and the user is not yet logged in.

TABLE 3–1 Use Case: Authenticate Portal User *(Continued)*

Item	Description
Description	1. User enters the portal URL. 2. If the customization parameter [remember login] is set, then automatically login the user and provide a session ID. 3. If first time user, prompt for LDAP user ID and password. 4. User enters previously assigned user ID and password. 5. Information is passed to Access Manager for validation. 6. If authentication passes, assign session ID and continue. 7. If authentication fails, display error message, return user to login page; decrement remaining attempts; if preset attempts exceed limit, notify user and lock out the account.

Quality of Service Requirements

To provide a quality of service, address the following topics:

- "Performance" on page 44
- "Availability" on page 45
- "Scalability" on page 45
- "Security" on page 46

Performance

Work with portal system administrators and portal developers to set the portal performance objectives based on the projected requirements of your portal. Objectives include the number of users, the number of concurrent users at peak load time and their usage pattern in accessing Portal Server.

You need to determine these two factors:

- Are you tuning for portal applications rapid response?
- Are you tuning for a large number of user concurrency?

As the number of users concurrently connected to the portal increase, the response time decreases given the same hardware and same set of parameters. Hence, gather information about the level of usage expected on your Portal Server, the anticipated number of concurrent users at any given time, the number of Portal Desktop activity requests, the amount of portal channel usage, acceptable response time for the end-user which is determined by your organization, and an optimal hardware configuration to meet the criteria.

Availability

Making Portal Server highly available involves ensuring high availability on each of the following components:

- **Portal Server**. In open mode, you can use a load balancer to detect a failed server component and redirect requests to other servers. In secure mode, Gateway components can detect the presence of a failed server component and redirect requests to other servers. (This is valid as long as the web container is the Sun Java System Web Server.)
- **Gateway**. A load balancer used with the Gateway detects a failed Gateway component and routes new requests to other Gateways. A load balancer also has the ability to intelligently distribute the workload across the server pool. Routing is restored when the failed Gateway recovers. Gateway components are stateless (session information is stored on the client in an HTTP cookie) so rerouting around a failed Gateway is transparent to users.
- **Directory Server**. A number of options make the LDAP directory highly available.
- **Netlet and Rewriter Proxies**. In the case of a software crash, a watchdog process automatically restarts the proxies. In addition, the Gateway performs load balancing and failure detection failover for the proxies.

Scalability

Scalability is a system's ability to accommodate a growing user population, without performance degradation, by the addition of processing resources. The two general means of scaling a system are vertical and horizontal scaling. The subject of this section is the application of scaling techniques for the Portal Server.

Benefits of scalable systems include:

- Improved response time
- Fault tolerance
- Manageability
- Expendability
- Simplified application development
- Building modules

Vertical Scaling

In *vertical scaling*, CPUs, memory, multiple instances of Portal Server, or other resources are added to one machine. This enables more process instances to run simultaneously. In Portal Server, you want to make use of this by planning and sizing to the number of CPUs you need.

Horizontal Scaling

In *horizontal scaling*, machines are added. This also enables multiple simultaneous processing and a distributed work load. In Portal Server, you make use of horizontal scaling because you can run the

Portal Server, Directory Server and Access Manager on different nodes. Horizontal scaling can also make use of vertical scaling, by adding more CPUs, for example.

Additionally, you can scale a Portal Server installation horizontally by installing server component instances on multiple machines. Each installed server component instance executes an HTTP process, which listens on a TCP/IP port whose number is determined at installation time. Gateway components use a round-robin algorithm to assign new session requests to server instances. While a session is established, an HTTP cookie, stored on the client, indicates the session server. All subsequent requests go to that server.

The section “Designing for Localization” on page 48, discusses an approach to a specific type of configuration that provides optimum performance and horizontal scalability.

Security

Security is the set of hardware, software, practices, and technologies that protect a server and its users from malicious outsiders. In that regard, security protects against unexpected behavior.

You need to address security globally and include people and processes as well as products and technologies. Unfortunately, too many organizations rely solely on firewall technology as their only security strategy. These organizations do not realize that many attacks come from employees, not outsiders. Therefore, you need to consider additional tools and processes when creating a secure portal environment.

Operating Portal Server in a secure environment involves making certain changes to the Solaris™ Operating Environment, the Gateway and server configuration, the installation of firewalls, and user authentication through Directory Server and SSO through Access Manager. In addition, you can use certificates, SSL encryption, and group and domain access.

Access Control

Portal Server relies on the HTTPS encryption protocol, in addition to UNIX system security, for protecting the Portal Server system software.

Security is provided by the web container, which you can configure to use SSL, if desired. Portal Server also supports SSL for authentication and end-user registration. By enabling SSL certificates on the web server, the Portal Desktop and other web applications can also be accessed securely. You can use the Access Manager policy to enforce URL-based access policy.

Portal Server depends on the authentication service provided by Access Manager and supports single sign-on (SSO) with any product that also uses the Access Manager SSO mechanism. The SSO mechanism uses encoded cookies to maintain session state.

Another layer of security is provided by SRA. It uses HTTPS by default for connecting the client browser to the intranet. The Gateway uses the Rewriter or Proxylet to enable all intranet web sites to be accessed without exposing them directly to the Internet. The Gateway also provides URL-based access policy enforcement without having to modify the web servers being accessed.

Communication from the Gateway to the server and intranet resources can be HTTPS or HTTP. Communication within the Portal Server system, for example between web applications and the directory server, does not use encryption by default, but it can be configured to use SSL.

UNIX User Installation

You can install and configure Portal Server to run under three different UNIX users:

- `root`. This is the default option. All Portal Server components are installed and configured to run as the system superuser. Some security implications arise from this configuration: An application bug can be exploited to gain root access tothe system. You need root access to modify some of the templates. This raises potential security ocncerns as theis responsibility is typically delegated to non-system administrators who can pose a threat to the system.
- **User** `nobody`. You can install Portal Server as the user `nobody` (uid 60001). This can improve the security of the system, because the user `nobody` does not have any privileges and cannot create, read, or modify the system files. This feature prevents user `nobody` from using Portal Server to gain access to system files and break into the system.

  The user `nobody` does not have a password, which prevents a regular user from becoming nobody. Only the superuser can change users without being prompted for a password. Thus, you still need `root` access to start and stop Portal Server services.
- **Non-**`root` **user**. You can run Portal Server as a regular UNIX user. The security benefits of a regular user are similar to the security benefits provided by the user `nobody`. A regular UNIX user has additional benefits as this type of user can start, stop, and configure services. After installation, you need to change ownership of some files.

  See the *Sun Java Enterprise System 2005Q5 Installation Guide* for more information.

Limiting Access Control

While the traditional security UNIX model is typically viewed as all-or-nothing, you can use alternative tools to provide some additional flexibility. These tools provide the mechanisms needed to create a fine grain access control to individual resources, such as different UNIX commands. For example, this toolset enables Portal Server to be run as `root`, while allowing certain users and roles superuser privileges to start, stop, and maintain the Portal Server framework.

These tools include:

- **Role-Based Access Control (RBAC)**. Solaris 8 and higher include the Role-Based Access Control (RBAC) to package superuser privileges and assign them to user accounts. RBAC enables separation of powers, controlled delegation of privileged operations to users, and a variable degree of access control.
- **Sudo.** Sudo is publicly available software, which enables a system administrator to give certain users the ability to execute a command as another user. See:

  `http://www.courtesan.com/sudo/sudo.html`

Using a Secure Access Zone

For maximum security, the Gateway is installed between two firewalls. The outermost firewall enables only SSL traffic from the Internet to the Gateways, which then direct traffic to servers on the internal network.

Designing for Localization

Localization is the process of adapting text and cultural content to a specific audience. Localization can be approached in two different ways:

- Localization of the entire product into a language that we don’t provide. This is usually done by a professional service organization.
- Localization of parts of Portal Server that can be translated to support localization include:
  - Template and JSP files
  - Resource Bundles
  - Display profile properties

For advanced language localization, create a well-defined directory structure for template directories. To preserve the upgrade path, maintain custom content and code outside of default directories. See the *Sun Java System Portal Server 7 2006Q1 Developer’s Guide* for more information on localization.

CHAPTER 4

Logical Design

During the logical design phase of the solution life cycle, you design a logical architecture showing the interrelationships of the logical components of the solution. The logical architecture and the usage analysis from the technical requirements phase form a deployment scenario, which is the input to the deployment design phase.

This chapter contains the following sections:

- "About Logical Architectures" on page 49
- "Portal Server Components" on page 51
- "Secure Remote Access Components" on page 52
- "Portal Server Nodes" on page 62
- "Example Portal Server Logical Architectures" on page 66
- "Example SRA Logical Architectures" on page 78

About Logical Architectures

A logical architecture identifies the software components needed to implement a solution, showing the interrelationships among the components. The logical architecture and the quality of service requirements determined during the technical requirements phase form a deployment scenario. The deployment scenario is the basis for designing the deployment architecture, which occurs in the next phase, deployment design.

Analysis of Logical Architecture

A high-level logical architecture provides the basis for a low-level logical architecture. The high-level logical architecture needs to meet the business and technical needs that you previously established. The logical architecture is broken down according to the various applications that comprise the system as a whole and the way in which users interact with it. In general, the logical architecture includes Portal Server Secure Remote Access (SRA), high availability, security (including Access Manager, and Directory Server architectural components.

The high- and low-level architectures also need to account for any factors beyond the control of the portal, including your network, hardware failures, and improper channel design.

The low-level architecture specifies such items as the physical architecture, network infrastructure, Portal Desktop channel and container design and the actual hardware and software components.

High-Level Logical Architecture

The high-level logical architecture to supports both the business and technical requirements and addresses questions such as:

- Does the proposed architecture support both the business and technical requirements?
- Can any modifications strengthen this architecture?
- Are there alternative architectures that might accomplish this?
- What is the physical layout of the system?
- What is the mapping of various components and connectivity?
- What is the logical definition describing the different categories of users and the systems and applications users have access to?
- Does the design account for adding more hardware to the system as required by the increase in web traffic over time?

Low-Level Logical Architecture

Low-level architecture focuses on specifying the processes and standards you use to build your portal solution, and specifying the actual hardware and software components of the solution, including:

- The Portal Server complex of servers.
- Network connectivity, describing how the portal complex attaches to the "outside world." Within this topic, you need to take into account security issues, protocols, speeds, and connections to other applications or remote sites.
- Information architecture, including user interfaces, content presentation and organization, data sources, and feeds.
- Access Manager architecture, including the strategy and design of organizations, suborganizations, roles, groups, and users, which is critical to long-term success.
- Integration strategy, including how the portal acts as an integration point for consolidating and integrating various information, and bringing people together in new ways.

Portal Server Components

Portal Server deployment consists of the following components:

- Sun Java™ System Access Manager

  Access Manager provides user and service management, authentication and single sign-on services, policy management, logging service, debug utility, the administration console, and client support interfaces for Portal Server. This consists of:

  - Java Development Kit (JDK™)

    Java Development Kit software provides the Java run-time environment for all Java software in Portal Server and its underlying components. Portal Server depends on the JDK software in the web container.

  - Network Security Services for Java software
  - Sun Java System Web Server
  - Java API for XML Processing (JAXP)
- Sun Java System Directory Server

  Directory Server provides the primary configuration and user profile data repository for Portal Server. The Directory Server is LDAP compliant and implemented on an extensible, open schema.

- Web Containers
  - Sun Java System Web Server
  - Sun Java System Application Server Enterprise Edition

    The following web containers can be used in place of the Web Server and Application Server software:

  - BEA WebLogic Server
  - IBM WebSphere® Application Server

    See the *Sun Java Enterprise System 2005Q5 Installation Guide* for information on deploying Portal Server in various web containers.

Note – See the latest Portal Server Release Notes for specific versions of products supported by Portal Server.

In addition to the components that make up the portal, your design should include (but is not limited to) the following:

- Contents from RDBMs
- Third-party content providers
- Custom developed providers and content
- Integration with back-end systems such as messaging and calendaring systems

- Role of the Content Management System
- Customer Resource Management
- Whether the portal runs in open or secure mode (requires Secure Remote Access)
- Usage estimates, which include your assumptions on the total number of registered users, average percentage of registered users logged in per day, average concurrent users that are logged in per day, average login time, average number of content channels that a logged in user has selected, and average number of application channels that a logged in user has selected.

Additionally, you need to consider how the following three network zones fit into your design:

- **Internet.** The public Internet is any network outside of the intranet and DMZ. Users portal server and securely access the Gateway and from here.
- **Demilitarized Zone (DMZ).** A secure area between two firewalls, enabling access to internal resources while limiting potential for unauthorized entry. The Gateway resides here where it can securely direct traffic from the application and content servers to the Internet.
- **Intranet.** Contains all resource servers. This includes intranet applications, web content servers, and application servers. The Portal Server and Directory Server reside here.

The logical architecture also describes the Portal Desktop look and feel, including potential items such as:

- Default page, with its default banner, logo, channels; total page weight, that is, total number of bytes of all the components of the page, including HTML, style sheet, JavaScript™, and image files; total number of HTTP requests for the page, that is, how many HTTP requests are required to complete downloading the page.
- Personalized pages, with channels that users can conceivably display and what preferences are available.

The logical architecture is where you also develop a caching strategy, if your site requires one. If the pages returned to your users contain references to large numbers of images, Portal Server can deliver these images for all users. However, if these types of requests can be offloaded to a reverse proxy type of caching appliance, you can free up system resources so that Portal Server can service additional users. Additionally, by placing a caching appliance closer to end users, these images can be delivered to end users somewhat more quickly, thus enhancing the overall end user experience.

Secure Remote Access Components

This section describes the following SRA components:

- "SRA Gateway" on page 53
- "Netlet" on page 55
- "Netlet Proxy" on page 57
- "NetFile" on page 57
- "Rewriter" on page 60
- "Rewriter Proxy" on page 61

- “Proxylet” on page 61

SRA Gateway

The SRA Gateway is a standalone Java process that can be considered to be stateless, since state information can be rebuilt transparently to the end user. The Gateway listens on configured ports to accept HTTP and HTTPS requests. Upon receiving a request, the Gateway checks session validity and header information to determine the type of request. Depending on the type of request, the Gateway performs the following:

- **Netlet request**. Routes the request (traffic) to the server specified in the Netlet rule that the user clicked in the Portal Desktop.
- **HTTP(S) traffic**. Routes the request to the server as specified by the HTTP header. Upon receiving a response from the server, the Gateway translates the response so that all intranet links within the response work on the extranet.

All the Gateway configuration information is stored in the Access Manager’s LDAP database as a profile. A gateway profile consists of all the configuration information related to the Gateway except.

All machine-specific information, such as machine-specific information such as host name and IP address, is stored in a configuration file in the local file system where the Gateway is installed. This enables one gateway profile to be shared between Gateways that are running on multiple machines.

As mentioned previously, you can configure the Gateway to run in both HTTP and HTTPS modes, simultaneously. This helps both intranet and extranet users to access the same Gateway: extranet users over HTTPS, and intranet users over HTTP (without the overhead of SSL).

Multiple Gateway Instances

If desired, you can run multiple Gateway instances on a single machine—this is referred as a multihomed Gateway. Each Gateway instance listens on separate port(s). You can configure Gateway instances to contact the same Portal Server instance, or different Portal Server instances. When running multiple instances of a Gateway on the same machine, you can associate an independent certificate database with each instance of the Gateway, and bind that Gateway to a domain. In essence, this provides the flexibility of having a different Gateway server certificate for each domain.

Multiple Portal Server Instances

Note – Session stickiness is not required in front of a Gateway (unless you are using Netlet), however performance is improved with session stickiness. On the other hand, session stickiness to the Portal Server instances is enforced by SRA.

Proxies

The Gateway uses proxies that are specified in its profile to retrieve contents from various web servers within the intranet and extranet. You can dedicate proxies for hosts and DNS subdomains and domains. Depending on the proxy configuration, the Gateway uses the appropriate proxy to fetch the required contents. If the proxy requires authentication, the proxy name is stored as part of the gateway profile, that the Gateway uses automatically, when connecting to the proxy.

Gateway and HTTP Basic Authentication

The Gateway supports basic authentication, that is, prompting for a user ID and password but not protecting those credentials during transmission from the user's computer to the site's web server. Such protection usually requires the establishment of a secure HTTP connection, typically through the use of SSL.

If a web server requires basic authentication the client prompts for user name and password and sends the information back to the requesting server. With the Gateway enabled for HTTP basic authentication, it captures the user name and password information and stores a copy in the user's profile in the Access Manager for subsequent authentications and login attempts. The original data is passed by the Gateway to the destination web server for basic authentication. The web server performs the validation of the user name and password.

The Gateway also enables fine control of denying and allowing this capability on an individual host basis.

Gateway and SSL Support

The Gateway supports both SSL v2 and SSL v3 encryption while running in HTTPS mode. You can use the Portal Server administration console to enable or disable specific encryption. The Gateway also supports Transport Layer Security (TLS).

SSL v3 has two authentication modes:

- **Mandatory server authentication**. The client must authenticate the server.
- **Optional authentication**. The server is configured to authenticate the client.

Personal Digital Certificate (PDC) authentication is a mechanism that authenticates a user through SSL client authentication. The Gateway supports PDC authentication with the support of Access Manager authentication modules. With SSL client authentication, the SSL handshake ends at the Gateway. This PDC-based authentication is integrated along with the Access Manager's certificate-based authentication. Thus, the client certificate is handled by Access Manager and not by the Gateway.

If the session information is not found as part of the HTTP or HTTPS request, the Gateway directly takes the user to the authentication page by obtaining the login URL from Access Manager. Similarly, if the Gateway finds that the session is not valid as part of a request, it takes the user to the login URL and at successful login, takes the user to the requested destination.

After the SSL session has been established, the Gateway continues to receive the incoming requests, checks session validity, and then forwards the request to the destination web server.

The Gateway server handles all Netlet traffic. If an incoming client request is Netlet traffic, the Gateway checks for session validity, decrypts the traffic, and forwards it to the application server. If Netlet Proxy is enabled, the Gateway checks for session validity and forwards it to Netlet Proxy. The Netlet Proxy then decrypts and forwards it to the application server.

Note – Because 40-bit encryption is very insecure, the Gateway provides an option that enables you to reject connections from a 40-bit encryption browser.

Gateway Access Control

The Gateway enforces access control by using Allowed URLs and Denied URLs lists. Even when URL access is allowed, the Gateway checks the validly of the session against the Access Manager session server. URLs that are designated in the Non Authenticated URL list bypass session validation, as well as the Allowed and Denied lists. Entries in the Denied URLs list take precedence over entries in the Allowed URLs list. If a particular URL is not part of any list, then access is denied to that URL. The wildcard character, *, can also be used as a part of the URL in either the Allow or Deny list.

Gateway Logging

You can monitor the complete user behavior by enabling logging on the Gateway. The Gateway uses the Portal Server logging API for creating logs.

Using Accelerators with the Gateway

You can configure accelerators, which are dedicated hardware co-processors, to off-load the SSL functions from a server's CPU. Using accelerators frees the CPU to perform other tasks and increases the processing speed for SSL transactions.

Netlet

Netlet can provide secure access to fixed port applications and some dynamic port applications that are available on the intranet from outside the intranet. The client can be behind a remote firewall and SSL proxy, or directly connected to the Internet. All the secure connections made from outside the intranet to the intranet applications through the Netlet are controlled by Netlet rules.

A Netlet applet running on the browser sets up an encrypted TCP/IP tunnel between the remote client machine and intranet applications on the remote hosts. Netlet listens to and accepts connections on preconfigured ports, and routes both incoming and outgoing traffic between the client and the destination server. Both incoming and outgoing traffic is encrypted using an encryption algorithm selected by the user, or configured by the administrator. The Netlet rule contains the details of all servers, ports, and encryption algorithms used in a connection. Administrators create Netlet rules by using the Portal Server administration console.

Static and Dynamic Port Applications

Static port applications run on known or static ports. Examples include IMAP and POP servers, Telnet daemons, and jCIFS. For static port applications, the Netlet rule includes the destination server port so that requests can be routed directly to their destinations.

Dynamic applications agree upon a port for communication as part of the handshake. You can include the destination server port as part of the Netlet rule. The Netlet needs to understand the protocol and examine the data to find the port being used between the client and the server. FTP is a dynamic port application. In FTP, the port for actual data transfer between the client and server is specified through the `PORT` command. In this case, the Netlet parses the traffic to obtain the data channel port dynamically.

Currently, FTP and Microsoft Exchange are the only dynamic port applications that Portal Server supports.

Note – Although Microsoft Exchange 2000 is supported with Netlet, the following constraints apply:

- For Portal Server versions before 6.3:
  - You must configure Exchange to use STATIC ports.
  - Netlet does not work with Windows 2000 and XP because Windows 2000 and XP clients reserve the Exchange port (port 135) for the RPC Portmapper, which Active Directory uses. Previous versions of Windows did not reserve this port. Because the port is reserved, you cannot assign Netlet to it, and thus the port cannot provide the necessary tunneling.
  - The Outlook 2000 client has the limitation that it does not enable you to change the port on which you want to connect to the Exchange server.
- For Portal Server 6.3 and later versions, Proxylet technology was introduced for use with OWA and Sun Java Enterprise Server, Portal Server Secure Remote Access deployments issues. Portal Server Administrators should consider this technology for a better user experience.

Netlet and Application Integration

Netlet works with many third parties such as Graphon, Citrix, and pcAnywhere. Each of these products provides secure access to the user's Portal Desktop from a remote machine using Netlet.

Split Tunneling

Split tunneling allows a VPN client to connect to both secure sites and non-secure sites, without having to connect or disconnect the VPN—in this case, the Netlet—connection. The client determines whether to send the information over the encrypted path, or to send it by using the non-encrypted path. The concern over split tunneling is that you could have a direct connection from the non-secure Internet to your VPN-secured network, via the client. Turning off split tunneling (not allowing both connections simultaneously) reduces the vulnerability of the VPN (or in the case of Netlet) connection to Internet intrusion.

Though Portal Server does not prohibit nor shut down multiple network connections while attached to the portal site, it does prevent unauthorized users from “piggybacking” on other users’s sessions in the following ways:

- Netlet is an application specific VPN and not a general purpose IP router. Netlet only forwards packets that have been defined by a Netlet rule. This differs from the standard VPN approach that gives you complete LAN access once you’ve connected to the network.
- Only an authenticated portal user can run the Netlet. No portal application can be run until the user has been successfully authenticated, and no new connections can be made if an authenticated session does not exist.
- All access controls in place on the application side are still in effect so that an attacker would also have to break in to the back-end application.
- Every Netlet connection results in a dialog box posted by the Netlet (running in the authenticated user’s JVM™) to the authenticated user’s display. The dialog box asks for verification and acknowledgement to permit the new connection. For attackers to be able to utilize a Netlet connection, attackers would need to know that the Netlet was running, the port number it was listening on, how to break the back-end application, and convince the user to approve the connection.

Netlet Proxy

A Netlet Proxy helps reduce the number of open ports needed in the firewall to connect the Gateway and the destination hosts.

For example, consider a configuration where users need Netlet to connect with a large number of Telnet, FTP, and Microsoft Exchange servers within the intranet. Assume that the Gateway is in a DMZ. If it routes the traffic to all the destination servers, a large number of ports would need to be open in the second firewall. To alleviate this problem, you can use a Netlet Proxy behind the second firewall and configure the Gateway to forward the traffic to the Netlet Proxy. The Netlet Proxy then routes all the traffic to the destination servers in the intranet and you reduce the number of open ports required in the second firewall. You can also deploy multiple Netlet Proxies behind the second firewall to avoid a single point of failure.

You could also use a third-party proxy to use only one port in the second firewall.

Note – Installing the Netlet Proxy on a separate node can help with Portal Server response time by offloading Netlet traffic to a separate node.

NetFile

NetFile enables remote access and operation of file systems that reside within the corporate intranet in a secure manner.

NetFile uses standard protocols such as NFS, jCIFS, and FTP to connect to any of the UNIX or Windows file systems that are permissible for the user to access. NetFile enables most file operations that are typical to file manager applications.

Components

To provide access to various file systems, NetFile has three components:

- **NetFile Java 1 Applet**. Has an AWT-based user interface. For use with older browsers that cannot support Java 2.
- **NetFile Java 2 Applet**. Has a Swing-based user interface. For use with browsers that support Java plug-ins.
- **NetFile servlet(s)**. Two NetFile servlets are present in the web container, one for each kind of NetFile applet. The servlets are responsible for connecting to different types of file systems, carrying out the operations that NetFile is configured to handle, and sending the information back to the applets for display.

NetFile is internationalized and provides access to file systems irrespective of their locale (character encoding).

NetFile uses Access Manager to store its own profile, as well as user settings and preferences. You administer NetFile through the Portal Server administration console.

Initialization

When a user selects a NetFile link in the Portal Server Desktop, the NetFile servlet checks if the user has a valid SSO token and permission to execute NetFile. If so, the applet is rendered to the browser. The NetFile applet connects back to the servlet to get its own configuration such as size, locale, resource bundle, as well as user settings and preferences. NetFile obtains the locale information and other user information (such as user name, mail ID, and mail server) using the user's SSO token. The user settings include any settings that the user has inherited from an organization or role, settings that are customized by the user, and settings that the user has stored upon exit from a previous NetFile session.

Validating Credentials

NetFile uses the credentials supplied by users to authenticate users before granting access to the file systems.

The credentials include a user name, password, and Windows or Novell domain (wherever applicable). Each share can have an independent password, therefore, users need to enter their credentials for every share (except for common hosts) that you add.

NetFile uses UNIX Authentication from the Access Manager to grant access to NFS file systems. For file systems that are accessed over FTP and jCIFs protocols, NetFile uses the methods provided by the protocol itself to validate the credentials.

Access Control

NetFile provides various means of file system access control. You can deny access to users to a particular file system based on the protocol. For example, you can deny a particular user, role, or organization access to file systems that are accessible only over NFS.

You can configure NetFile to allow or deny access to file systems at any level, from organization, to suborganization, to user. You can also allow or deny access to specific servers. Access can be allowed or denied to file systems for users depending on the type of host, including Windows, FTP, NFS, and FTP over NetWare. For example, you can deny access for Windows hosts to all users of an organization. You can also specify a set of common hosts at an organization or role level, so that all users in that organization or role can access the common hosts without having to add them for each and every member of the organization or role.

As part of the NetFile service, you can configure the Allowed URLs or Denied URLs lists to allow or deny access to servers at the organization, role, or user level. The Denied URLs list takes precedence over the Allowed URLs. The Allowed URLs and Denied URLs lists can contain the * wildcard to allow or deny access to a set of servers under a single domain or subdomain.

Security

When you use NetFile with SRA configured for SSL, all connections made from NetFile applets to the underlying file system happen over the SSL connection established between the Gateway and the browser. Because you typically install the Gateway in a DMZ, and open a limited number of ports (usually only one) in the second firewall, you do not compromise security while providing access to the file systems.

Special Operations

NetFile is much like a typical file manager application with a set of features that are appropriate for a remote file manager application. NetFile enables users to upload and download files between the local and remote file systems (shares). You can limit the size of the upload file (from the local to the remote file system) through the Portal Server administration console.

NetFile also enables users to select multiple files and compress them by using GZIP and ZIP compression. Users can select multiple files and send them in a single email as multiple attachments. NetFile also uses the SSO token of Access Manager to access the user’s email settings (such as IMAP server, user name, password, and reply-to address) for sending email.

Double-clicking a file in the NetFile window launches the application corresponding to the MIME type and opens the file. NetFile provides a default MIME types configuration file that has mappings for most popular file types (extensions) and MIME-types that you can edit for adding new mappings.

You can search for files and display the list in a separate window using NetFile. The results of each search are displayed in a new window while maintaining the previous search result windows. The type of character encoding to be used for a particular share is user configurable, and is part of the share’s setting. If no character encoding is specified, NetFile uses ISO-8859-1 while working with the shares. The ISO-8859-1 encoding is capable of handling most common languages. ISO-8859-1 encoding gives NetFile the capability to list files in any language and to transferring files in any language without damaging the file contents.

NetFile creates temporary files only when mailing files (in both NetFile Java 1 and Java 2). Temporary files are not created during uploading and downloading files between Windows file systems and the local file systems over the jCIFS protocol.

Note – NetFile supports deletion of directories and remote files. All the contents of remote directories are deleted recursively.

NetFile and Multithreading

NetFile uses multithreading to provide the flexibility of running multiple operations simultaneously. For example, users can launch a search operation, start uploading files, then send files by using email. NetFile performs all three operations simultaneously and still permit the user to browse through the file listing.

Rewriter

Rewriter is an independent component that translates all URIs (in both HTML and JavaScript code) to ensure that the intranet content is always fetched through the Gateway. You define a ruleset (a collection of rules) that identifies all URLs that need to be rewritten in a page. The ruleset is an XML fragment that is written according to a Document Type Definition (DTD). Using the generic ruleset that ships with the Rewriter, you can rewrite most URLs (but not all) without any additional rules. You can also associate rulesets with domains for domain-based translations.

An external ruleset identifies the URI in the content. Any request that needs to be served by SRA follows this route:

▼ Route for SRA Requests

1. **From the request, SRA identifies the URI of the intranet page or Internet page that needs to be served.**

2. **SRA uses the proxy settings to connect to the identified URI.**

3. **The domain of the URI is used to identify the ruleset to be used to rewrite this content.**

4. **After fetching the content and ruleset, SRA inputs these to the Rewriter where identified URIs are translated.**

5. **The original URI is replaced with the rewritten URI.**

6 **This process is repeated until the end of the document is reached.**

7 **The resultant Rewriter output is routed to the browser.**

Rewriter Proxy

To minimize the number of open ports in the firewall, use the Rewriter Proxy. When you install the Rewriter Proxy, HTTP requests are redirected to the Rewriter Proxy instead of directly to the destination host. The Rewriter Proxy in turn sends the request to the destination server.

Using the Rewriter Proxy enables secure HTTP traffic between the Gateway and intranet computers and offers two advantages:

- If a firewall is between the Gateway and server, the firewall needs to open only two ports. One firewall is between the Gateway and the Rewriter Proxy and another is between the Gateway and the Portal Server.
- You can use a third-party proxy to use only one port in the second firewall to read the Rewriter Proxy.
- HTTP traffic is now secure between the Gateway and the intranet even if the destination server only supports HTTP protocol (not HTTPS).

Note – You can run multiple Rewriter Proxies to avoid a single point of failure and achieve load balancing.

Proxylet

Proxylet is a dynamic proxy server that runs on a client machine. Proxylet redirects a URL to the Gateway. It does this by reading and modifying the proxy settings of the browser on the client machine so that the settings point to the local proxy server or Proxylet.

It supports both HTTP and SSL, inheriting the transport mode from the Gateway. If the Gateway is configured to run on SSL, Proxylet establishes a secure channel between the client machine and the Gateway. Proxylet uses the Java 2 Enterprise Edition API if the client JVM is 1.4 or higher or if the required jar files reside on the client machine. Otherwise it uses the KSSL API.

Proxylet is enabled from the Portal Server administration console where the client IP address and port are specified.

Unlike Rewriter, Proxylet is an out-of-the-box solution with very little or no post-installation changes. Also Gateway performance improves because Proxylet does not deal with web content.

Portal Server Nodes

Usually, but not always, you deploy Portal Server software on the following different portal nodes (servers) that work together to implement the portal:

- **Portal Server node**. The web server where Portal Server resides. You can also install the Search component on this node if desired. Access Manager can reside here.
- **Access Manager node**.The server where Access Manager can reside. Access Manager does not have to reside on the same node as Portal Server.
- **Search node.** Optional. The server you use for the Portal Server Search service. You can install the Portal Server Search service on its own server for performance, scalability and availability reasons.
- Gateway nodes. Optional. The server where the SRA Gateway resides. You can install the Gateway on the portal node. Because you locate the Gateway in the DMZ, the Gateway is installed on a separate, non-portal node.
- Netlet Proxy node. Optional. The server used to run applications securely between users' remote desktops and the servers running applications on your intranet.
- **Rewriter Proxy node**. Optional. The server used to run applications securely between users' remote desktops and the servers running applications on your intranet.
- **Directory Server node**. The server running Directory Server software. You can install Directory Server on a non-portal node.
- **Other servers**. These servers, such as mail, file, and legacy servers, provide backend support, data, and applications to portal users.

Portal Server and Access Manager on Different Nodes

Portal Server and Access Manager can be located on different nodes. This type of deployment provides the following advantages:

- Identity services can be deployed separately from portal services. Portal Server can be one of many applications using identity services.
- Authentication and policy services can be separate from provider applications including Portal Server related applications.
- Access Manager can be used by other web containers to assist with development of portal customizations.

Note – When Portal Server and Access Manager are on different nodes, the Access Manager SDK must reside on the same node as Portal Server. The web application and supporting authentication daemons can reside on a separate node from the Portal Server instance.

The Access Manager SDK consists of the following components:

Identity Management SDK–provides the framework to create and manage users, roles, groups, containers, organizations, organizational units, and sub-organizations.

Authentication API and SPI–provides remote access to the full capabilities of the Authentication Service.

Utility API–manages system resources.

Loggin API and SPI–records, among other things, access approvals, access denials and user activity.

Client Detection API–detects the type of client browser that is attempting to access its resources and respond with the appropriately formatted pages.

SSO API–provides interfaces for validating and managing session tokens, and for maintaining the user's authentication credentials.

Policy API–evaluates and manages Access Manager policies and provides additional functionality for the Policy Service.

SAML API–exchanges acts of authentication, authorization decisions and attribute information.

Federation Management API–adds functionality based on the Liberty Alliance Project specifications.

Portal Server System Communication Links

Figure 4–1 shows the processes and communication links of a Portal Server system that are critical to the availability of the solution.

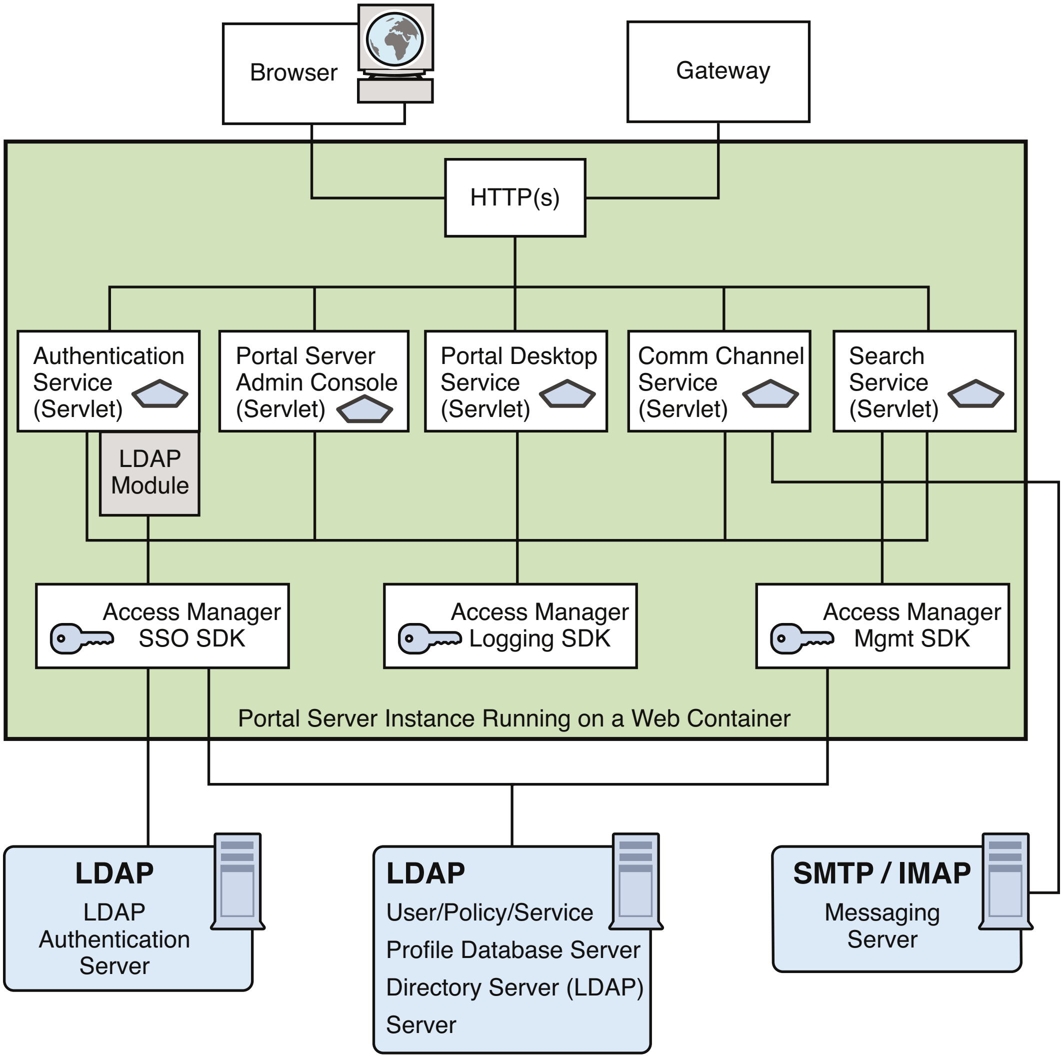
Browser
Gateway
HTTP(s)
Authentication Service (Servlet)
Portal Server Admin Console (Servlet)
Portal Desktop Service (Servlet)
Comm Channel Service (Servlet)
Search Service (Servlet)
LDAP Module
Access Manager SSO SDK
Access Manager Logging SDK
Access Manager Mgmt SDK
Portal Server Instance Running on a Web Container
LDAP
LDAP Authentication Server
LDAP
User/Policy/Service
Profile Database Server
Directory Server (LDAP)
Server
SMTP / IMAP
Messaging Server

FIGURE 4–1 Portal Server Communication Links

In this figure, the box encloses the Portal Server instance running on Web Server technology. Within the instance are five servlets (Authentication, Portal Server administration console, Portal Desktop, Communication Channel, and Search), and the three SDKs (Access Manager SSO, Access Manager Logging, and Access Manager Management). The Authentication service servlet also makes use of an LDAP service provider module.

A user uses either a browser or the Gateway to communicate with Portal Server. This traffic is directed to the appropriate servlet. Communication occurs between the Authentication service's LDAP module and the LDAP authentication server; between the Communications channel servlet and the SMTP/IMAP messaging server; between the Access Manager SSO SDK and the LDAP server; and between the Access Manager Management SDK and the LDAP server.

Figure 4–1 shows that if the following processes or communication links fail, the portal solution becomes unavailable to end users:

- **Portal Server Instance**. Runs in the context of a web container. Components within an instance communicate through the JVM using Java APIs. An instance is a fully qualified domain name and a TCP port number. Portal Server services are web applications that are implemented as servlets or JSP™ files

  Portal Server is built on top of Access Manager for authentication single sign-on (session) management, policy, and profile database access. Thus, Portal Server inherits all the benefits (and constraints) of Access Manager with respect to availability and fault tolerance.

  By design, Access Manager's services are either stateless or the services can share context data. Services can recover to the previous state in case of a service failure.

  Within Portal Server, Portal Desktop does not share state data among instances. This means that an instance redirect causes the user context to be rebuilt for the enabled services. Usually, redirected users do not notice this because Portal Server services can rebuild a user context from the user's profile, and by using contextual data stored in the request. While this statement is generally true for out-of-the-box services, it might not be true for channels or custom code. Developers need to be careful to not design stateful channels to avoid loss of context upon instance failover.
- **Profile Database Server**. The profile database server is implemented by Directory Server software. Although this server is not strictly part of Portal Server, availability of the server and integrity of the database are fundamental to the availability of the system.
- **Authentication Server**. This is the directory server for LDAP authentication (usually, the same server as the profile database server). You can apply the same high availability techniques to this server as for the profile database server.
- **SRA Gateway and Proxies** . The SRA Gateway is a standalone Java technology process that can be considered stateless, because state information can be rebuilt transparently to end users. The Gateway profile maintains a list of Portal Server instances and does round robin load balancing across the Gateway instances. Session stickiness is not required in front of a Gateway, but with session stickiness, performance is better. On the other hand, session stickiness to Portal Server instances is enforced by SRA.

  SRA includes other Java technology processes called Netlet Proxy and Rewriter Proxy. You use these proxies to extend the security perimeter from behind the firewall, and limit the number of holes in the DMZ. You can install these proxies on separate nodes.

Example Portal Server Logical Architectures

This section provides some examples of logical architectures for Portal Server:

- "A Typical Portal Server Installation" on page 66
- "Portal Server Building Modules" on page 67
- "Access Manager and Portal Server on Separate Nodes" on page 75
- "Two Portal Servers One Access Manager" on page 76
- "Two Portal Servers Two Access Managers" on page 76
- "One Load Balancer Two Access Managers" on page 77

A Typical Portal Server Installation

Figure 4–2 illustrates some of the components of a portal deployment but does not address the actual physical network design, single points of failure, nor high availability.

This illustration shows the high-level architecture of a typical installation at a company site for a business-to-employee portal. In this figure, the Gateway is hosted in the company's DMZ along with other systems accessible from the Internet, including proxy/cache servers, web servers, and mail Gateways. The portal node, portal search node, and directory server, are hosted on the internal network where users have access to systems and services ranging from individual employee desktop systems to legacy systems.

Note – If you are designing an ISP hosting deployment, which hosts separate Portal Server instances for business customers who each want their own portal, contact your Sun representative. Portal Server requires customizations to provide ISP hosting functionality.

Figure 4–2 shows users on the Internet accessing the Gateway from a browser. The Gateway connects the user to the IP address and port for portal users attempting to access. For example, a B2B portal would usually allow access to only port 443, the HTTPS port. Depending on the authorized use, the Gateway forwards requests to the portal node, or directly to the service on the enterprise internal network.

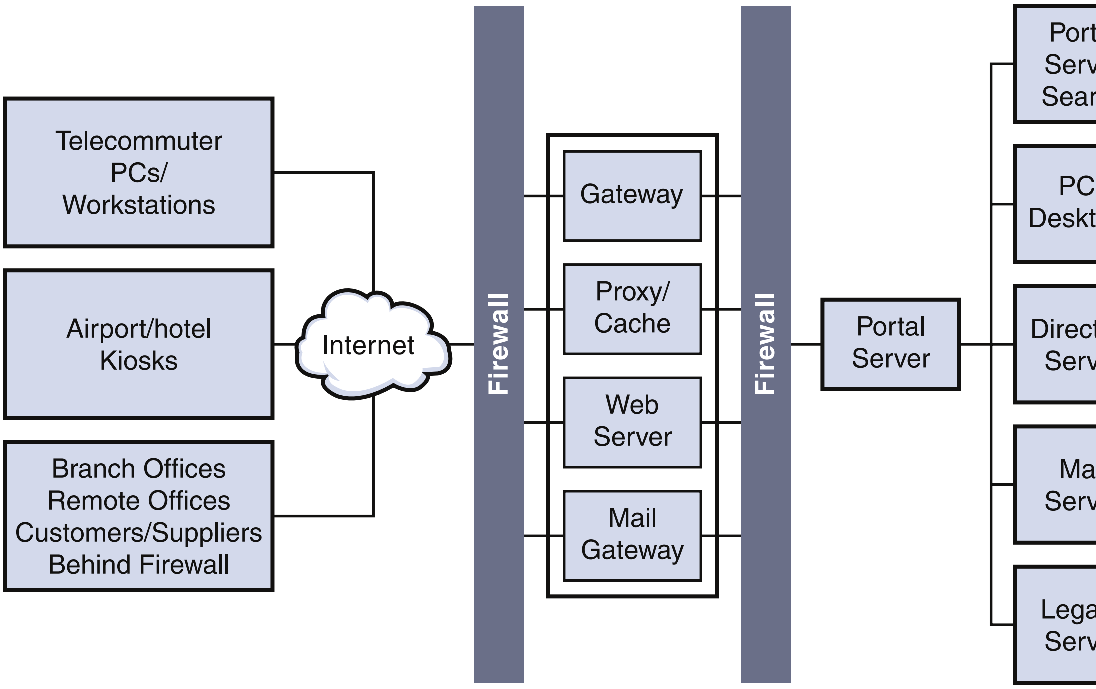

FIGURE 4–2 High-level Architecture for a Business-to-Employee Portal

Figure 4–3 shows a Portal Server deployment with SRA services.

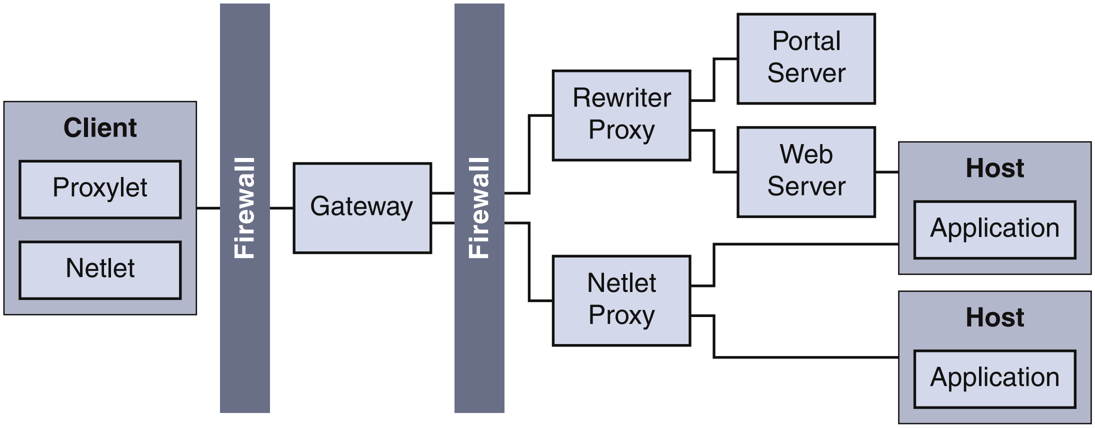

FIGURE 4–3 SRA Deployment

Portal Server Building Modules

Because deploying Portal Server is a complex process involving many other systems, this section describes a specific configuration that provides optimum performance and horizontal scalability. This configuration is known as a Portal Server *building module*.

A Portal Server building module is a hardware and software construct with limited or no dependencies on shared services. A typical deployment uses multiple building modules to achieve optimum performance and horizontal scalability. Figure 4–4 shows the building module architecture.

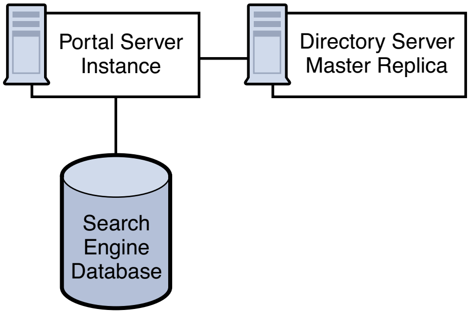

FIGURE 4–4 Portal Server Building Module Architecture

Note – The Portal Server building module is simply a recommended configuration. In some cases, a different configuration might result in slightly better throughput (usually at the cost of added complexity). For example, adding another instance of Portal Server to a four CPU system might result in up to ten percent additional throughput, at the cost of requiring a load balancer even when using just a single system.

Building Modules and High Availability Scenarios

Portal Server provides three scenarios for high availability:

- "Best Effort" on page 69

  The system is available as long as the hardware does not fail and as long as the Portal Server processes can be restarted by the watchdog process.

- "No Single Point of Failure" on page 70

  The use of hardware and software replication creates a deployment with no single point of failure (NSPOF). The system is always available, as long as no more than one failure occurs consecutively anywhere in the chain of components. However, in the case of failures, user sessions are lost.

- "Transparent Failover" on page 72

> The system is always available but in addition to NSPOF, failover to a backup instance occurs transparently to end users. In most cases, users do not notice that they have been redirected to a different node or instance. Sessions are preserved across nodes so that users do not have to reauthenticate. Portal Server services are stateless or use checkpointing mechanisms to rebuild the current execution context up to a certain point.

Possible supported architectures include the following:

- Using Sun™ Cluster software on components that support Sun Cluster agents
- Multi-master Directory Server techniques

This section explains implementing these architectures and leverages the building module concept, from a high-availability standpoint.

Table 4–1 summarizes these high availability scenarios along with their supporting techniques.

TABLE 4–1 Portal Server High Availability Scenarios

Component Requirements	Necessary for Best Effort Deployment?	Necessary for NSPOF Deployment?	Necessary for Transparent Failover Deployment?
Hardware Redundancy	Yes	Yes	Yes
Portal Server Building Modules	No	Yes	Yes
Multi-master Configuration	No	Yes	Yes
Load Balancing	Yes	Yes	Yes
Stateless Applications and Checkpointing Mechanisms	No	No	Yes
Session Failover	No	No	Yes.
Directory Server Clustering	No	No	Yes

Note – Load balancing is not provided out-of-the-box with the Web Server product.

Best Effort

In this scenario, you install Portal Server and Directory Server on a single node that has a secured hardware configuration for continuous availability, such as Sun Fire UltraSPARC™ III machines. (Securing a Solaris™ Operating Environment system requires that changes be made to its default configuration.)

This type of server features full hardware redundancy, including: redundant power supplies, fans, system controllers; dynamic reconfiguration; CPU hot-plug; online upgrades; and disks rack that can be configured in RAID 0+1 (striping plus mirroring), or RAID 5 using a volume management system, which prevents loss of data in case of a disk crash. Figure 4–5 shows a small, best effort deployment using the building module architecture.

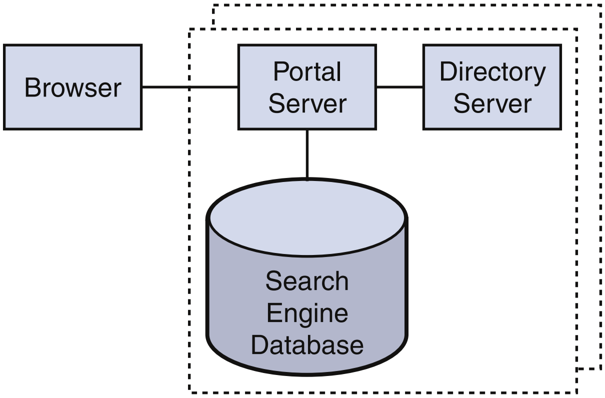

FIGURE 4–5 Best Effort Scenario

In this scenario, for memory allocation, four CPUs by eight GB RAM (4x8) of memory is sufficient for one building module. The Portal Server console is outside of the building module so that it can be shared with other resources. (Your actual sizing calculations might result in a different allocation amount.)

This scenario might suffice for task critical requirements. Its major weakness is that a maintenance action necessitating a system shutdown results in service interruption.

When SRA is used, and a software crash occurs, a watchdog process automatically restarts the Gateway, Netlet Proxy, and Rewriter Proxy.

No Single Point of Failure

Portal Server natively supports the no single point of failure (NSPOF) scenario. NSPOF is built on top of the best effort scenario, and in addition, introduces replication and load balancing.

Figure 4–6 shows a building module consisting of a a Portal Server instance, a Directory Server replica for profile reads and a search engine database. As such, at least two building modules are necessary to achieve NSPOF, thereby providing a backup if one of the building modules fails. These building modules consist of four CPUs by eight GB RAM.

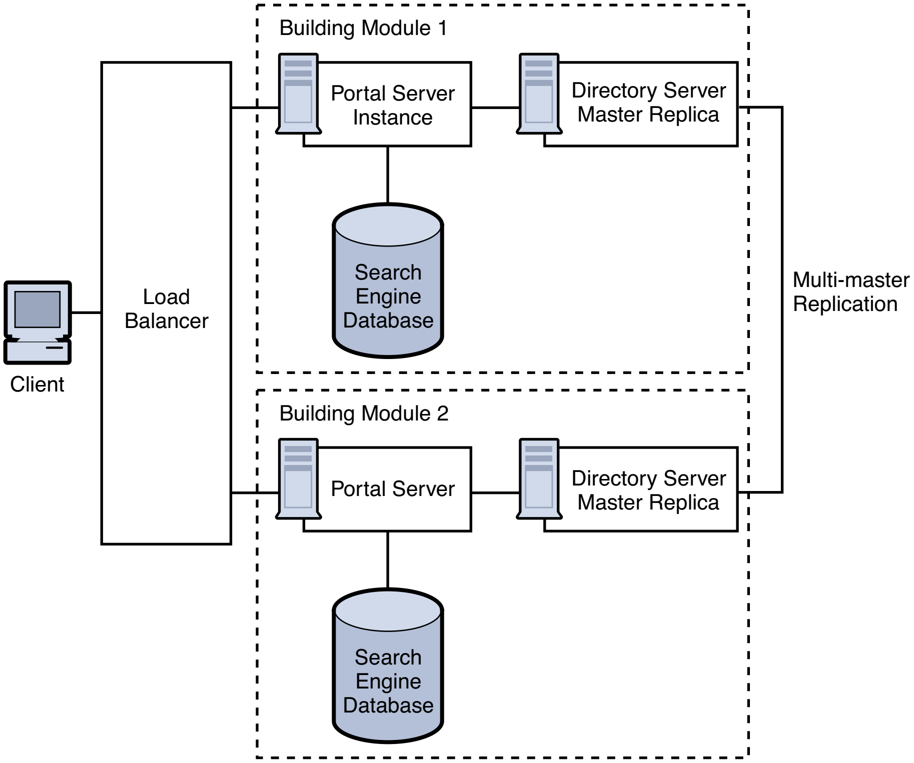

FIGURE 4–6 No Single Point of Failure Example

When the load balancer detects Portal Server failures, it redirects users' requests to a backup building module. Accuracy of failure detection varies among load balancing products. Some products are capable of checking the availability of a system by probing a service involving several functional areas of the server, such as the servlet engine, and the JVM. In particular, most vendor solutions from Resonate, Cisco, Alteon, and others enable you to create arbitrary scripts for server availability. As the load balancer is not part of the Portal Server software, you must acquire it separately from a third-party vendor.

Note – Access Manager requires that you set up load balancing to enforce sticky sessions. This means that once a session is created on a particular instance, the load balancer needs to always return to the same instance for that session. The load balancer achieves this by binding the session cookie with the instance name identification. In principle, that binding is reestablished when a failed instance is decommissioned. Sticky sessions are also recommended for performance reasons.

Multi-master replication (MMR) takes places between the building modules. The changes that occur on each directory are replicated to the other, which means that each directory plays both roles of supplier and consumer. For more information on MMR, refer to the *Sun Java System Directory Server Deployment Guide*.

Note – In general, the Directory Server instance in each building module is configured as a replica of a master directory, which runs elsewhere. However, nothing prevents you from using a master directory as part of the building module. The use of masters on dedicated nodes does not improve the availability of the solution. Use dedicated masters for performance reasons.

Redundancy is equally important to the directory master so that profile changes through the administration console or the Portal Desktop, along with consumer replication across building modules, can always be maintained. Portal Server and Access Manager support MMR. The NSPOF scenario uses a multi-master configuration. In this configuration, two suppliers can accept updates, synchronize with each other, and update all consumers. The consumers can refer update requests to both masters.

SRA follows the same replication and load balancing pattern as Portal Server to achieve NSPOF. As such, two SRA Gateways and pair of proxies are necessary in this scenario. The SRA Gateway detects a Portal Server instance failure when the instance does not respond to a request after a certain time-out value. When this occurs, the HTTPS request is routed to a backup server. The SRA Gateway performs a periodic check for availability until the first Portal Server instance is up again.

The NSPOF high availability scenario is suitable to business critical deployments. However, some high availability limitations in this scenario might not fulfill the requirements of a mission critical deployment.

Transparent Failover

Transparent failover uses the same replication model as the NSPOF scenario but provides additional high availability features, which make the failover to a backup server transparent to end users.

Figure 4–7 shows a transparent failover scenario. Two building modules are shown, consisting of four CPUs by eight GB RAM. Load balancing is responsible for detecting Portal Server failures and redirecting users' requests to a backup Portal Server in the building module. Building Module 1 stores sessions in the sessions repository. If a crash occurs, the application server retrieves sessions created by Building Module 1 from the sessions repository.

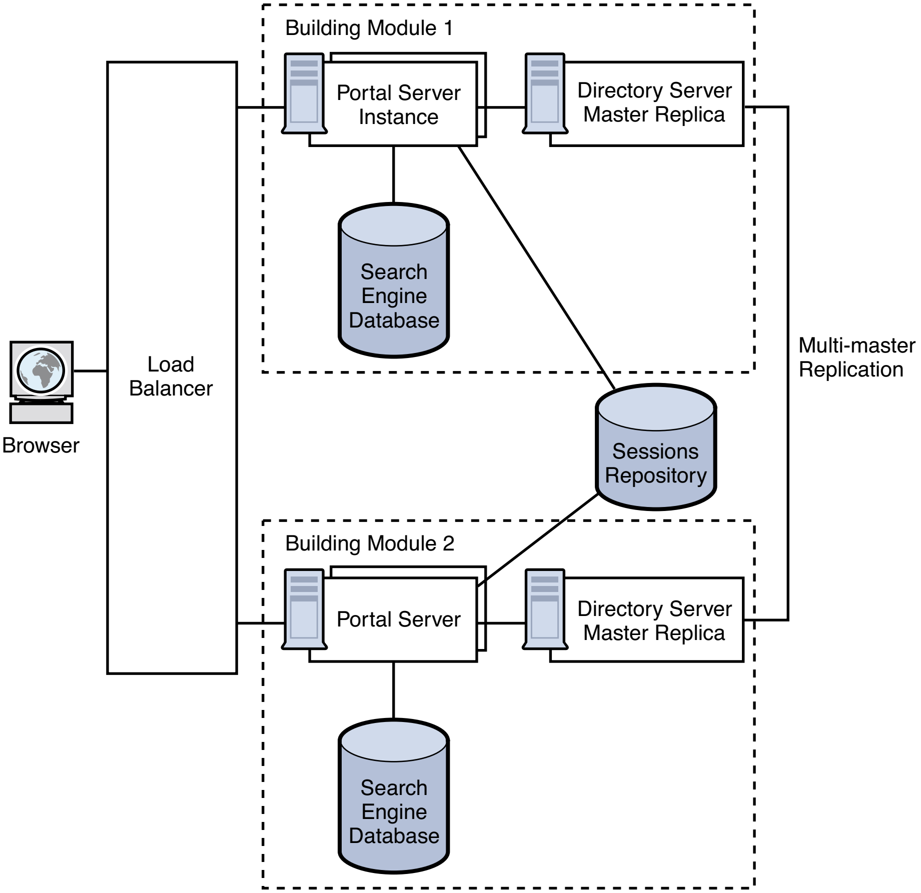

FIGURE 4–7 Transparent Failover Example Scenario

The session repository is provided by the application server software. Portal Server is running in an application server. Portal Server supports transparent failover on application servers that support HttpSession failover. See Chapter 9 for more information.

With session failover, users do not need to reauthenticate after a crash. In addition, portal applications can rely on session persistence to store context data used by the checkpointing. You configure session failover in the `AMConfig.properties` file by setting the `com.iplanet.am.session.failover.enabled` `property` to **`true`**.

The Netlet Proxy cannot support the transparent failover scenario because of the limitation of the TCP protocol. The Netlet Proxy tunnels TCP connections, and you cannot migrate an open TCP connection to another server. A Netlet Proxy crash drops off all outstanding connections that would have to be reestablished.

Building Module Solution Recommendations

This section describes guidelines for deploying your building module solution.

How you construct your building module affects performance. Consider the following recommendations to deploy your building module properly:

- Deploy a building module on a single machine.
- If you use multiple machines, or if your Portal Server machine is running a large number of instances, use a fast network interconnect.
- On servers with more than eight CPUs, create processor sets or domains with either two or four CPUs. For example, if you choose to install two instances of Portal Server on an eight CPU server, create two four-CPU processor sets.

Directory Server

Identify your Directory Server requirements for your building module deployment. For specific information on Directory Server deployment, see the *Directory Server Deployment Guide*.

Consider the following Directory Server guidelines when you plan your Portal Server deployment:

- The amount of needed CPU in the Directory Server consumer replica processor set depends on the number of Portal Server instances in the building module as well as performance and capacity considerations.
- If possible, dedicate a Directory Server instance for the sole use of the Portal Server instances in a building module.
- Map the entire directory database indexes and cache in memory to avoid disk latency issues.
- When deploying multiple building modules, use a multi-master configuration to work around bottlenecks caused by the profile updates and replication overhead to the Directory Server supplier.

LDAP

The scalability of building modules is based on the number of LDAP writes resulting from profile updates and the maximum size of the LDAP database.

Note – If the LDAP server crashes with the `_db` files in the `/tmp` directory, the files are lost when the server restarts. This improves performance but also affects availability.

If the analysis at your specific site indicates that the number of LDAP write operations is indeed a constraint, some of the possible solutions include creating building modules that replicate only a specific branch of the directory and a layer in front that directs incoming requests to the appropriate instance of portal.

Search Engine

When you deploy the Search Engine as part of your building module solution, consider the following:

- In each building module, make sure only one Portal Server instance has the Search Engine database containing the Resrouce Descriptors (RDs). The remaining Portal Server instances have default empty Search Engine databases.
- Factors that influence whether to use a building module for the portal Search database include the intensity of search activities in a Portal Server deployment, the range of search hits, and the average number of search hits for all users, in addition to the number of concurrent searches. For example, the load generated on a server by the Search Engine can be both memory and CPU intensive for a large index and heavy query load.
- You can install Search on a machine separate from Portal Server, to keep the main server dedicated to portal activity. When you do so, you use the `searchURL` property of the Search provider to point to the second machine where Search is installed. The Search instance is a normal portal instance. You install the Search instance just as you do the portal instance, but use it just for Search functionality.

Access Manager and Portal Server on Separate Nodes

Figure 4–8 illustrates Access Manager and Portal Server residing on separate nodes.

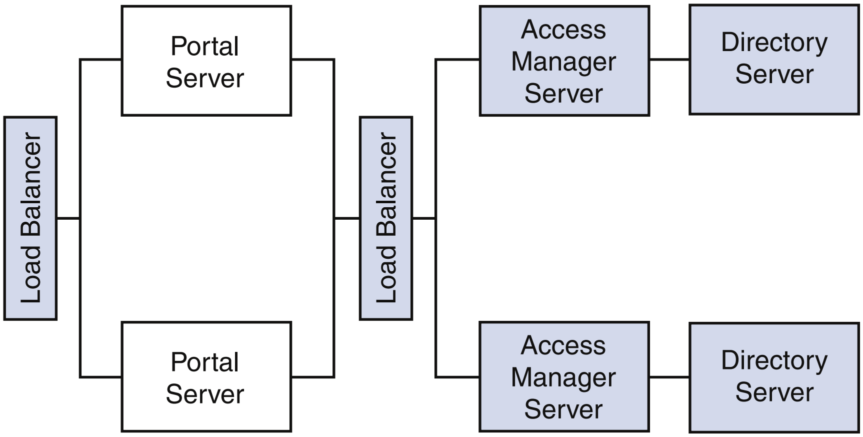

FIGURE 4–8 Access Manager and Portal Server on Different Nodes

As a result of this implementation of Portal Server and Access Manager separation, other topology permutations are possible for portal services architecture deployments as shown in the next three figures.

Two Portal Servers One Access Manager

Figure 4–9 shows two Portal Server instances configured to work with a single Access Manager and two Directory Servers where both the Access Manager and the Directory Servers operate in a Java Enterprise System Sun Clustered environment. This configuration is ideal when Access Manager and Directory Server instances are not the bottleneck.

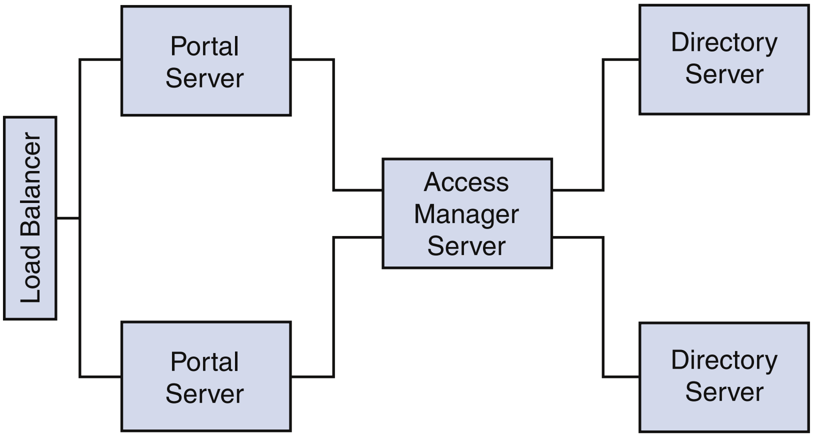

FIGURE 4–9 Two Portal Servers and One Access Manager

Two Portal Servers Two Access Managers

Figure 4–10 shows a configuration for maximum horizontal scalability and higher availability achieved by a horizontal server farm. Two Portals Servers can be fronted with a load balancer for maximum throughput and high availability.

Another load balancer can be put between Portal Servers and Access Managers to achieve authentication and policy processes as a load distributor and failover mechanism for higher availability.

In this scenario, Blade 1500s can be utilized for Portal Services to distribute the load, similar Blades can be used to host Access Manager Services and Directory Services respectively. With the architecture shown in Figure 4–10, a redundancy of services exists for each of the product stack, therefore, most of the unplanned downtime can be minimized or eliminated.

However, the planned downtime is still an issue. If an upgrade or patch includes changes to the Directory Server software schema used by the Access Manager software, all of the software components must be stopped to update the schema information stored in the Directory Server. However, updating schema information can be considered a fairly rare occurrence in most patch upgrades.

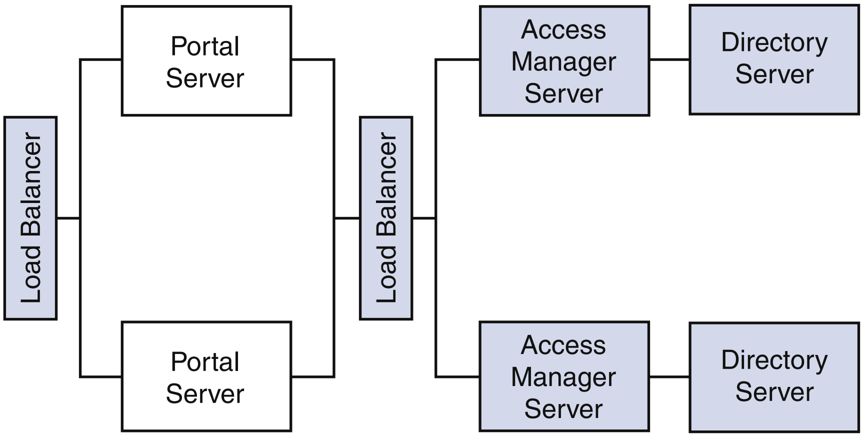

FIGURE 4–10 Two Portal Servers and Two Access Managers

One Load Balancer Two Access Managers

Figure 4–11 shows configuration allowing authentication throughput coming from Portal Server to be load-balanced across the two Access Managers.

This configuration could be implemented when the Portal Server resides on a high-end medium to large server (that is 1 to 4 processors) with a very wide bandwidth network connection. The Access Managers with the policy and authentication services could be on two medium-size servers.

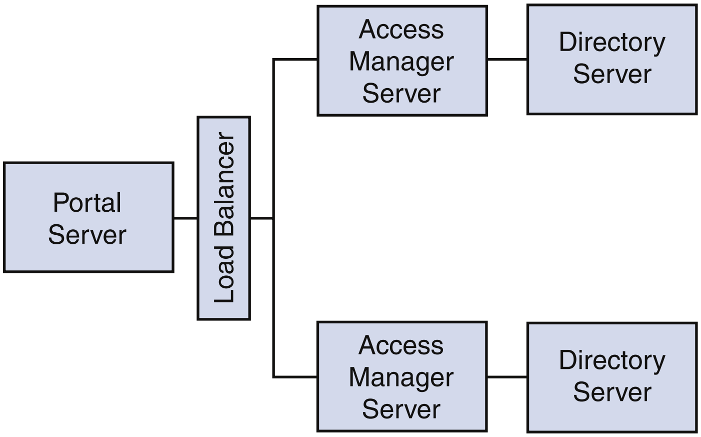

FIGURE 4–11 Load Balancing two Access Managers

Example SRA Logical Architectures

The SRA Gateway provides the interface and security barrier between the remote user sessions originating from the Internet and your organization's intranet. The Gateway serves two main functions:

- Provides basic authentication services to incoming user sessions, including establishing identity and allowing or denying access to the platform.
- Provides mapping and rewriting services to enable web-based links to the intranet content for users.

For Internet access, use 128-bit SSL to provide the best security arrangement and encryption or communication between the user's browser and Portal Server. The Gateway, Netlet, NetFile, Netlet Proxy, Rewriter Proxy, and Proxylet constitute the major components of SRA.

This section lists some of the possible configurations of these components. This section is meant only as a guide, not a complete deployment reference. Choose a configuration based on your business needs:

- "Basic SRA Configuration" on page 79
- "Disable Netlet" on page 80
- "Proxylet" on page 81
- "Multiple Gateway Instances" on page 82
- "Netlet and Rewriter Proxies" on page 83
- "Netlet and Rewriter Proxies on Separate Nodes" on page 85
- "Two Gateways and Netlet Proxy" on page 86
- "Gateway with Accelerator" on page 87
- "Netlet with 3rd Party Proxy" on page 88

- "Reverse Proxy" on page 89

Tip – To set up the authlessanonymous page to display through the Gateway, add `/portal/dt` to the non-authenticated URLs of the gateway profile. However, this means that even for normal users, portal pages will not need authentication and no session validation is performed.

Basic SRA Configuration

Figure 4–12 shows the most simple configuration possible for SRA. The figure shows a client browser running NetFile and Netlet. The Gateway is installed on a separate machine in the DMZ between two firewalls. The Portal Server is located on a machine beyond the second firewall in the intranet. The other application hosts that the client accesses are also located beyond the second firewall in the intranet.

The Gateway is in the DMZ with the external port open in the firewall through which the client browser communicates with the Gateway. In the second firewall, for HTTP or HTTPS traffic, the Gateway can communicate directly with internal hosts. If security policies do not permit it, use SRA proxies between the Gateway and the internal hosts. For Netlet traffic, the connection is direct from the Gateway to the destination host.

Without a SRA proxy, the SSL traffic is limited to the Gateway and the traffic is unencrypted from the Gateway to the internal host (unless the internal host is running in HTTPS mode). Any internal host to which the Gateway has to initiate a Netlet connection should be directly accessible from DMZ. This can be a potential security problem and hence this configuration is recommended only for the simplest of installations.

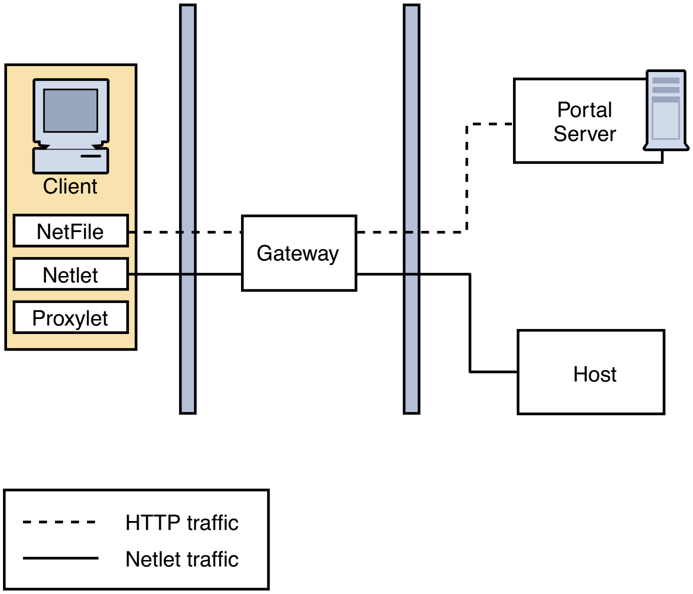

FIGURE 4–12 Basic SRA Configuration

Disable Netlet

Figure 4–13 shows a scenario similar to the basic SRA configuration except that Netlet is disabled. If the client deployment is not going to use Netlet for securely running applications that need to communicate with intranet, then use this setup for performance improvement.

You can extend this configuration and combine it with other deployment scenarios to provide better performance and a scalable solution.

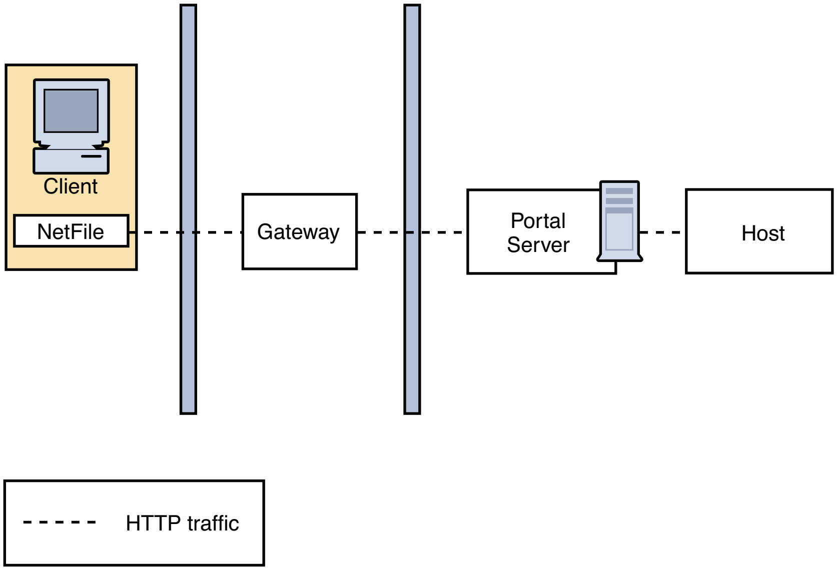

FIGURE 4–13 Disable Netlet

Proxylet

Figure 4–14 illustrates how Proxylet enables users to securely access intranet resources through the Internet without exposing these resources to the client.

It inherits the transport mode (either HTTP or HTTPS) from the Gateway.

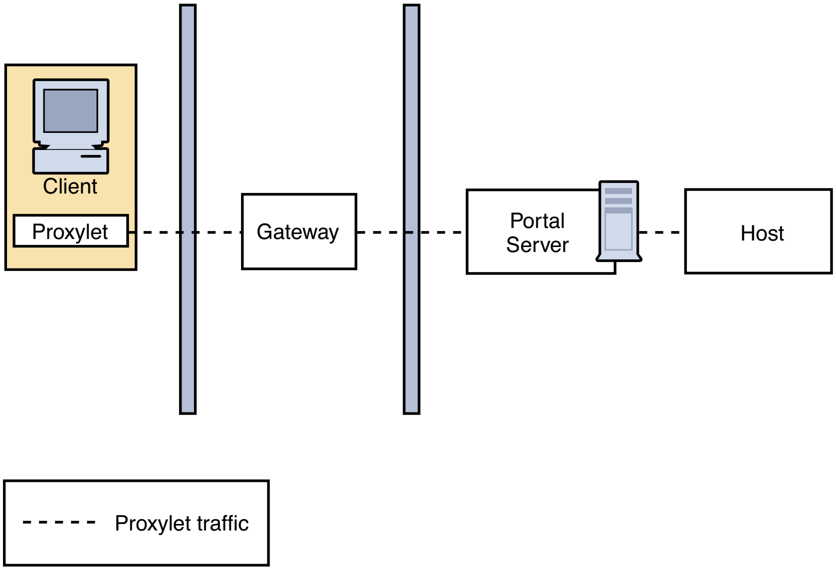

FIGURE 4–14 Proxylet

Multiple Gateway Instances

Figure 4–15 shows an extension of the SRA basic configuration. Multiple Gateway instances run on the same machine or multiple machines. You can start multiple Gateway instances with different profiles.

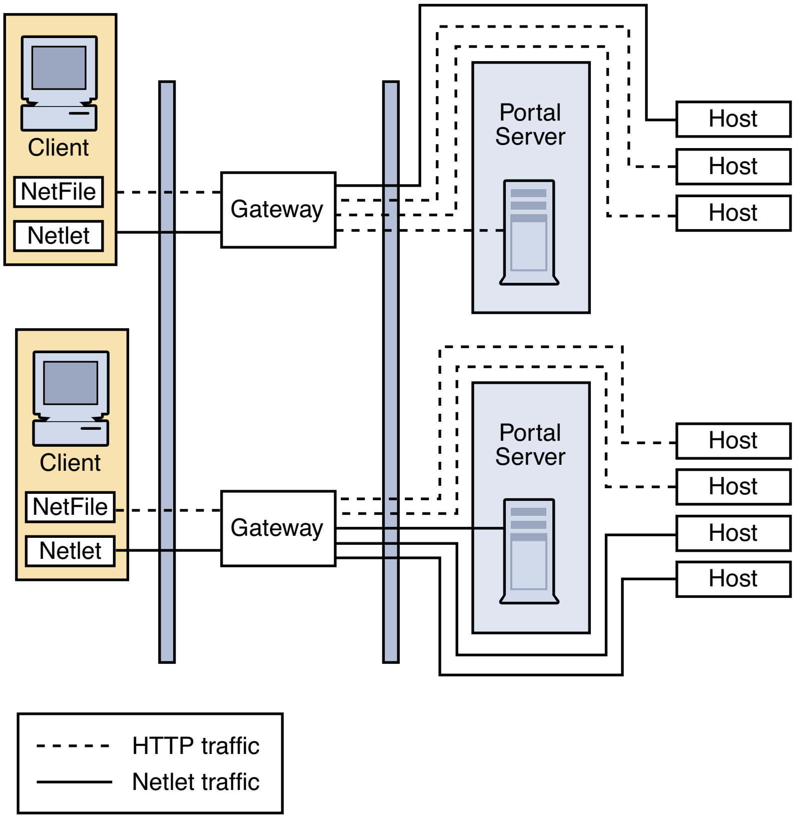

FIGURE 4–15 Multiple Gateway Instances

Note – Although Figure 4–15 shows a 1-to-1 correspondence between the Gateway and the Portal Servers, this need not necessarily be the case in a real deployment. You can have multiple Gateway instances, and multiple Portal Server instances, and any Gateway can contact any Portal Server depending on the configuration.

The disadvantage to this configuration is that multiple ports need to be opened in the second firewall for each connection request. This could cause potential security problems.

Netlet and Rewriter Proxies

Figure 4–16 shows a configuration with a Netlet Proxy and a Rewriter Proxy. With these proxies, only two open ports are necessary in the second firewall.

The Gateway need not contact the application hosts directly now, but will forward all Netlet traffic to the Netlet proxy and Rewriter traffic to the Rewriter Proxy. Since the Netlet Proxy is within the intranet, it can directly contact all the required application hosts without opening multiple ports in the second firewall.

The traffic between the Gateway in the DMZ and the Netlet Proxy is encrypted, and gets decrypted only at the Netlet Proxy, thereby enhancing security.

If the Rewriter Proxy is enabled, all traffic is directed through the Rewriter Proxy, irrespective of whether the request is for the Portal Server node or not. This ensures that the traffic from the Gateway in the DMZ to the intranet is always encrypted.

Because the Netlet Proxy, Rewriter Proxy, and Portal Server are all running on the same node, there might be performance issues in such a deployment scenario. This problem is overcome when proxies are installed on a separate nodes to reduce the load on the Portal Server node.

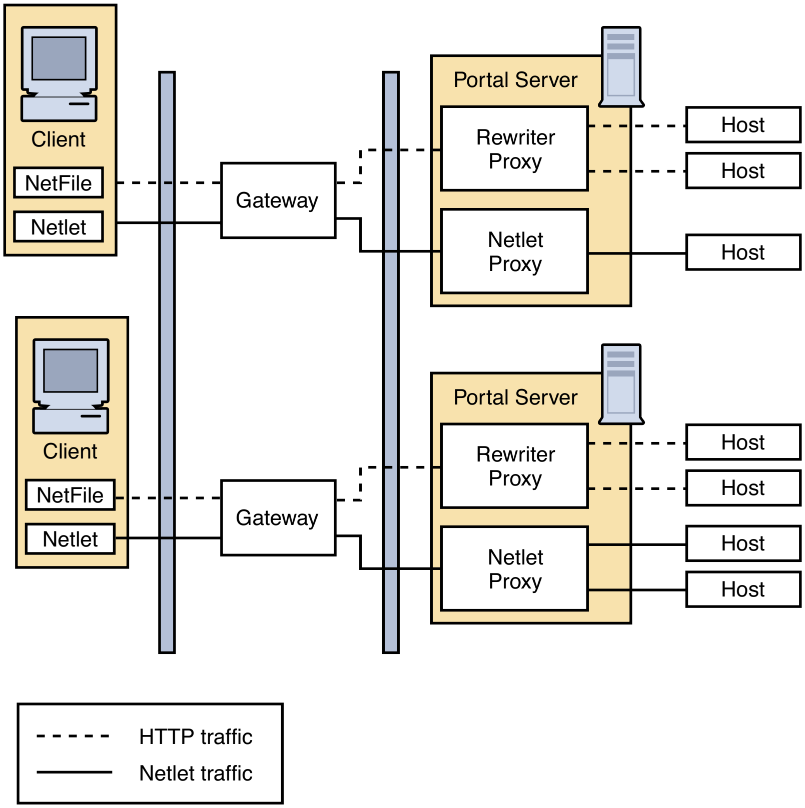

FIGURE 4–16 Netlet and Rewriter Proxies

Netlet and Rewriter Proxies on Separate Nodes

To reduce the load on the Portal Server node and still provide the same level of security at increased performance, you can install Netlet and Rewriter Proxies on separate nodes. This deployment has an added advantage in that you can use a proxy and shield the Portal Server from the DMZ. The node that runs these proxies needs to be directly accessible from the DMZ.

Figure 4–17 shows the Netlet Proxy and Rewriter Proxy on separate nodes. Traffic from the Gateway is directed to the separate node, which in turn directs the traffic through the proxies and to the required intranet hosts.

You can have multiple instances or installations of Netlet and Rewriter Proxies. You can configure each Gateway to try to contact various instances of the proxies in a round robin manner depending on availability.

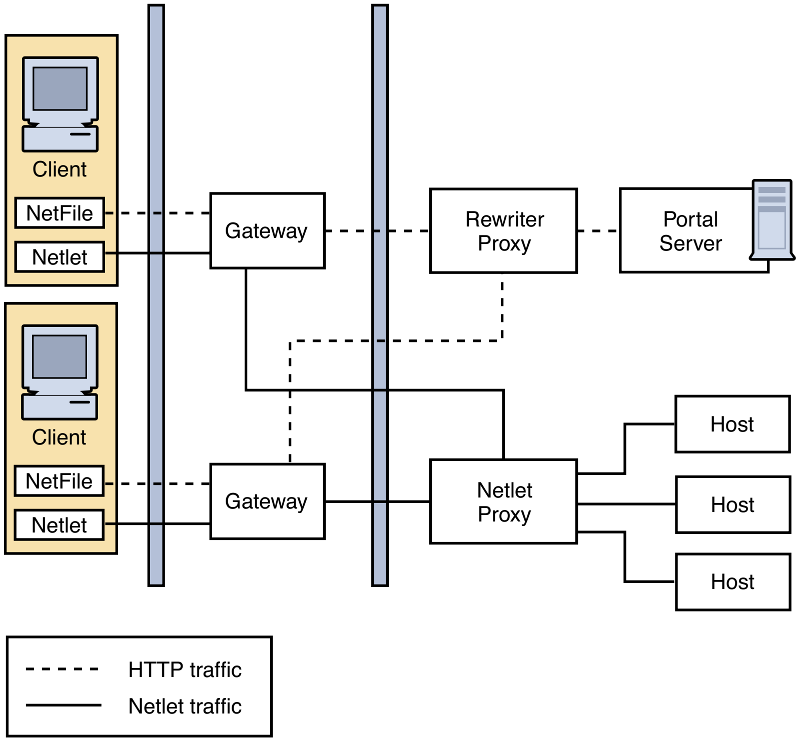

FIGURE 4–17 Proxies on Separate Nodes

Two Gateways and Netlet Proxy

Load balancers provide a failover mechanism for higher availability for redundancy of services on the Portal Servers and Access Managers.

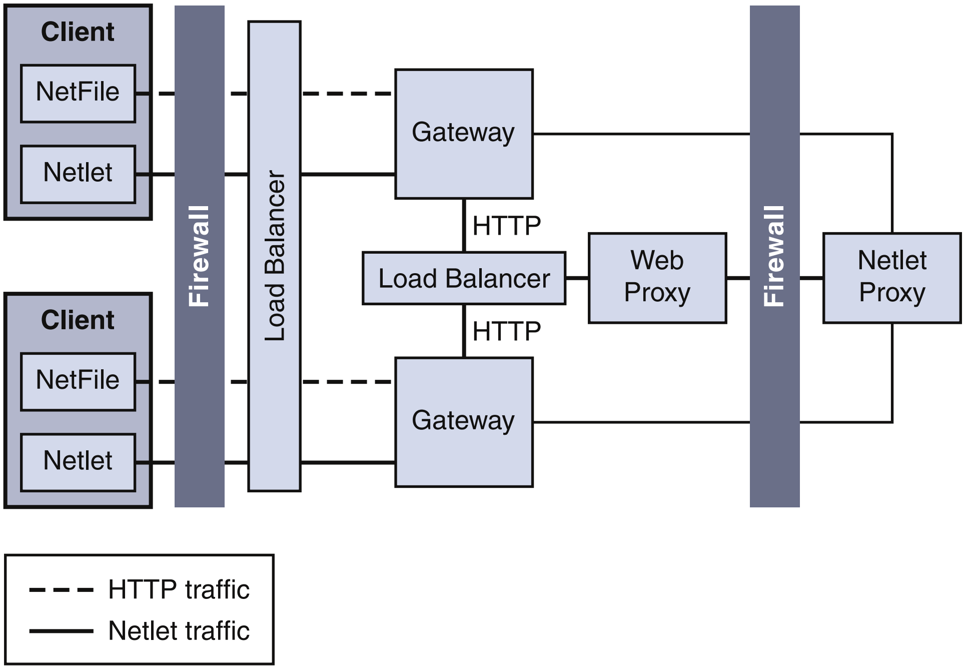

FIGURE 4–18 Two Gateways and Netlet Proxy

Gateway with Accelerator

You can configure an external SSL device to run in front of the Gateway in open mode. It provides the SSL link between the client and SRA. For information on accelerators, see the *Sun Java System Portal Server 6 Secure Remote Access 2005Q4 Administration Guide*

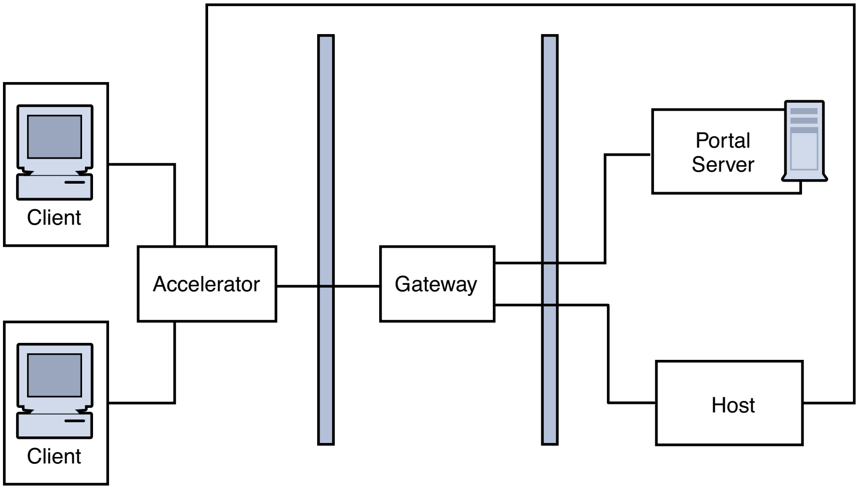

FIGURE 4–19 SRA Gateway with External Accelerator

Netlet with 3rd Party Proxy

“Netlet with 3rd Party Proxy” on page 88 illustrates using a third-party proxy to limit the number of ports in the second firewall to one. You can configure the Gateway to use a third-party proxy to reach the Rewriter and the Netlet Proxies.

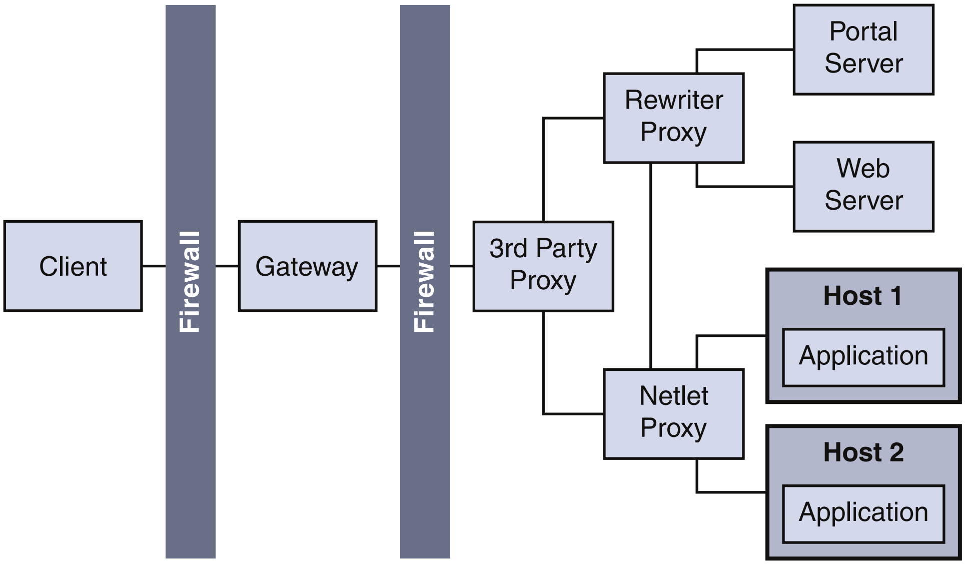

FIGURE 4–20 Netlet and Third-Party Proxy

Reverse Proxy

A proxy server serves Internet content to the intranet, while a reverse proxy serves intranet content to the Internet. Certain deployments of reverse proxy are configured to serve the Internet content to achieve load balancing and caching.

Figure 4–21 illustrates how you can configure a reverse proxy in front of the Gateway to serve both Internet and intranet content to authorized users. Whenever the Gateway serves web content, it needs to ensure that all subsequent browser requests based on this content are routed through the Gateway. This is achieved by identifying all URLs in this content and rewriting as appropriate.

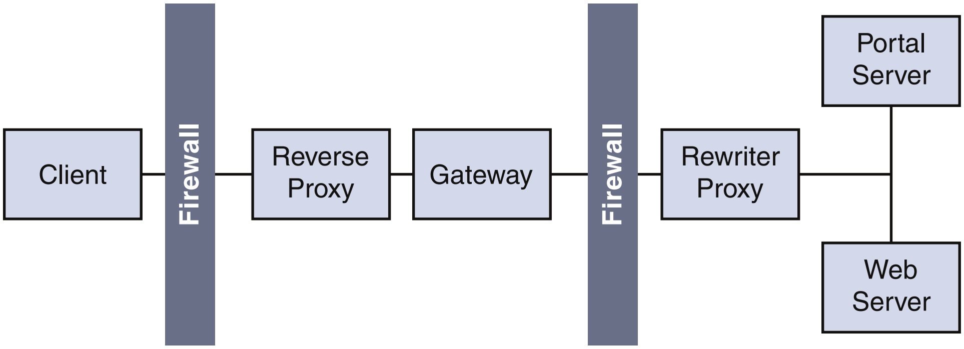

FIGURE 4–21 Using a Reverse Proxy in Front of the Gateway

Deployment Scenario

The completed logical architecture design by itself is not sufficient to move forward to the deployment design phase of the solution life cycle. You need to pair the logical architecture with the quality of service (QoS) requirements determined during the technical requirements phase. The pairing of the logical architecture with the QoS requirements constitutes a deployment scenario. The deployment scenario is the starting point for designing the deployment architecture, as explained in Chapter 5.

CHAPTER 5

Deployment Design

During the deployment design phase of the solution life cycle, you design a high-level deployment architecture and a low-level implementation specification, and prepare a series of plans and specifications necessary to implement the solution. Project approval occurs in the deployment design phase.

This chapter contains the following sections:

- "About Deployment Design" on page 91
- "Deployment Design Methodology" on page 92
- "Estimating Processor Requirements" on page 92
- "Identifying Performance Bottlenecks" on page 95
- "Optimizing Resources" on page 96
- "Managing Risks" on page 97

About Deployment Design

Deployment design begins with the deployment scenario created during the logical design and technical requirements phases of the solution life cycle. The deployment scenario contains a logical architecture and the quality of service (QoS) requirements for the solution. You map the components identified in the logical architecture across physical servers and other network devices to create a deployment architecture. The QoS requirements provide guidance on hardware configurations for performance, availability, scalability, and other related QoS specifications.

Designing the deployment architecture is an iterative process. You typically revisit the QoS requirements and reexamine your preliminary designs. You take into account the interrelationship of the QoS requirements, balancing the trade-offs and cost of ownership issues to arrive at an optimal solution that ultimately satisfies the business goals of the project.

Deployment Design Methodology

As with other aspects of deployment planning, deployment design is as much an art as it is a science and cannot be detailed with specific procedures and processes. Factors that contribute to successful deployment design are past design experience, knowledge of systems architecture, domain knowledge, and applied creative thinking.

Deployment design typically revolves around achieving performance requirements while meeting other QoS requirements. The strategies you use must balance the trade-offs of your design decisions to optimize the solution. The methodology you use typically involves the following tasks:

- **Estimating processsor requirements**. Deployment design often begins with portal sizing in the logical architecture. Start with the use cases and modify your estimates accordingly. Also consider any previous experience you have with designing enterprise systems.
- **Estimating processsor requirements for secure transport**. Study the use cases that require secure transport and modify CPU estimates accordingly.
- **Replicating services for availability and scalability**. Once you are satisfied with the processor estimates, make modifications to the design to account for QoS requirements for availability and scalability. Consider load balancing solutions that address availability and failover considerations.

  During your analysis, consider the trade-offs of your design decisions. For example, what affect does the availability and scalability strategy have on serviceability (maintenance) of the system? What are the others costs of the strategies?
- **Identifying bottlenecks.** As you continue with your analysis, examine the deployment design to identify any bottlenecks that cause the transmission of data to fall beneath requirements, and make adjustments.
- **Optimizing resources**. Review your deployment design for resource management and consider options that minimizes costs while fulfilling requirements.
- **Managing risks**. Revisit your business and technical analyses with respect to your design, making modifications to account for events or situations that might not have been foreseen in the earlier planning.

Estimating Processor Requirements

With a baseline figure established in the usage analysis, you can then validate and refine that figure to account for scalability, high availability, reliability, and good performance:

▼ Steps to Estimate Processor Requirements

1. “Customize the Baseline Sizing Figures” on page 93
2. “Validate Baseline Sizing Figures” on page 93

3. “Refine Baseline Sizing Figures” on page 94

4. “Validate Your Final Figures” on page 95

   The following sections describe these steps.

Customize the Baseline Sizing Figures

Establishing an appropriate sizing estimate for your Portal Server deployment is an iterative process. You might wish to change the inputs to generate a range of sizing results. Customizing your Portal Server deployment can greatly affect its performance.

After you have an estimate of your sizing, consider:

- “LDAP Transaction Numbers” on page 93
- “Application Server Requirements” on page 93

LDAP Transaction Numbers

Use the following LDAP transaction numbers for an out-of-the-box portal deployment to understand the impact of the service demand on the LDAP master and replicas. These numbers change once you begin customizing the system.

- Access to authless anonymous portal - 0 ops
- Login by using the Login channel - 2 BINDS, 2 SRCH
- Removing a channel from the Portal Desktop - 8 SRCH, 2 MOD
- Reloading the Portal Desktop - 0 ops

Application Server Requirements

One of the primary uses of Portal Server installed on an application server is to integrate portal providers with Enterprise JavaBeans™ architecture and other J2EE™ technology stack constructs, such as Java Database Connectivity (JDBC™) and J2EE Connector Architecture (JCA), running on the application server. These other applications and modules can consume resources and affect your portal sizing.

Validate Baseline Sizing Figures

Now that you have an estimate of the number of CPUs for your portal deployment, use a trial deployment to measure the performance of the portal. Use load balancing and stress tests to determine:

- Throughput, the amount of data processed in a specified amount of time
- Latency, the period of time that one component is waiting for another component
- Maximum number of concurrent sessions

Portal samples are provided with the Portal Server. You can use them, with channels similar to the ones you will use, to create a load on the system. The samples are located on the Portal Desktop.

Use a trial deployment to determine your final sizing estimates. A trial deployment helps you to size back-end integration to avoid potential bottlenecks with Portal Server operations.

Refine Baseline Sizing Figures

Your next step is to refine your sizing figure. In this section, you build in the appropriate amount of headroom so that you can deploy a portal site that features scalability, high availability, reliability and good performance.

Because your baseline sizing figure is based on so many estimates, do not use this figure without refining it.

When you refine your baseline sizing figure:

- Use your baseline sizing figure as a reference point.
- Expect variations from your baseline sizing figure.
- Learn from the experience of others.
- Use your own judgement and knowledge.
- Examine other factors in your deployment.

  If the Portal Server deployment involves multiple data centers on several continents and even traffic, you need a higher final sizing figure than if you have two single data centers on one continent with heavy traffic.
- Plan for changes.

  A portal site is likely to experience various changes after you launch it. Changes you might encounter include the following:

  - An increase in the number of channels
  - Growth in the user base
  - Modification of the portal site's purpose
  - Changes in security needs
  - Power failures
  - Maintenance demands

    Considering these factors enables you to develop a sizing figure that is flexible and enables you to avoid risk when your assumptions regarding your portal change following deployment.

    The resulting figure ensures that your portal site has:

  - Scalability high availability, reliability and high performance
  - Room for whatever you want to provide
  - Flexibility for adjusting to changes

Validate Your Final Figures

Use a trial deployment to verify that the portal deployment satisfies your business and technical requirements.

Identifying Performance Bottlenecks

Before reading the section on memmory consumption and it's affect on performance, read the following document on tuning garbage collection with the Java Virtual Machine, version 1.4.2:

`http://java.sun.com/products/hotspot/index.html`

Memory Consumption and Garbage Collection

Portal Server requires substantial amounts of memory to provide the highest possible throughput. At initialization, a maximum address space is virtually reserved but does not allocate physical memory unless needed. The complete address space reserved for object memory can be divided into the young and old generations.

Most applications suggest using a larger percentage of the total heap for the new generation, but in the case of Portal Server, using only one eighth the space for the young generation is appropriate, because most memory used by Portal Server is long-lived. The sooner the memory is copied to the old generation the better the garbage collection (GC) performance.

Even with a large heap size, after a portal instance has been running under moderate load for a few days, most of the heap appears to be used because of the lazy nature of the GC. The GC performs full garbage collections until the resident set size (RSS) reaches approximately 85 percent of the total heap space; at that point the garbage collections can have a measurable impact on performance.

For example, on a 900 MHz UltraSPARCIII™, a full GC on a 2 GB heap can take over ten seconds. During that period of time, the system is unavailable to respond to web requests. During a reliability test, full GCs are clearly visible as spikes in the response time. You must understand the impact on performance and the frequency of full GCs. In production, full GCs go unnoticed most of the time, but any monitoring scripts that measure the performance of the system need to account for the possibility that a full GC might occur.

Measuring the frequency of full GCs is sometimes the only way to determine if the system has a memory leak. Conduct an analysis that shows the expected frequency (of a baseline system) and compare that to the observed rate of full GCs. To record the frequency of GCs, use the `vebose:gc` JVM™ parameter.

Optimizing Resources

You can optimize resources by using the following:

- "Hardware Accelerator" on page 96
- "Sun Enterprise Midframe Line" on page 96

Hardware Accelerator

SSL-intensive servers, such as the SRA Gateway, require large amounts of processing power to perform the encryption required for each secure transaction. Using a hardware accelerator in the Gateway speeds up the execution of cryptographic algorithms, thereby increasing the performance speed.

The Sun Crypto Accelerator 1000 board is a short PCI board that functions as a cryptographic co-processor to accelerate public key and symmetric cryptography. This product has no external interfaces. The board communicates with the host through the internal PCI bus interface. The purpose of this board is to accelerate a variety of computationally intensive cryptographic algorithms for security protocols in e-commerce applications.

See the *Sun Java System Portal Server 6 Secure Remote Access 2005Q4 Administration Guide* for more information on the Sun Crypto Accelerator 1000 board and other accelerators.

Note – The Sun Crypto Accelerator 1000 board supports only SSL handshakes and not symmetric key algorithms. This is not generic to all other cryptographic accelerators. Other cryptographic accelerators are on the market and some of them can support symmetric key encryption.

You could use a hardware accelerator on the Netlet Proxy and Rewriter Proxy machine and derive some performance improvement.

Sun Enterprise Midframe Line

Normally, for a production environment, you would deploy Portal Server and SRA on separate machines. However, in the case of the Sun Enterprise™ midframe machines, which support multiple hardware domains, you can install both Portal Server and SRA in different domains on the same Sun Enterprise midframe machine. The normal CPU and memory requirements that pertain to Portal Server and SRA still apply; you would implement the requirements for each in the separate domains.

In this type of configuration, pay attention to security issues. For example, in most cases the Portal Server domain is located on the intranet, while the SRA domain is in the DMZ.

Managing Risks

This section contains a few tips to help you in the sizing process.

- A business-to-consumer portal requires that you deploy SRA to use the Gateway and SSL. Make sure you take this into account for your sizing requirements. Once you turn on SSL, the performance of the portal can be up to ten times slower than without SSL.
- For a business-to-employee portal, make sure that you have a user profile that serves as a baseline.
- For any portal, build in headroom for growth. This means not just sizing for today's needs, but future needs and capacity. This includes usual peaks after users return from a break, such as a weekend or holiday, or if usage is increased over time because the portal is more "sticky."
- If you are deploying your portal solution across multiple geographic sites, you need to fully understand the layout of your networks and data centers
- Decide what type of redundancy you need. Consider items such as production down time, upgrades, and maintenance work. In general, when you take a portal server out of production, the impact to your capacity should be no more than one quarter of the overall capacity.
- In general, usage concurrencies for a business-to-employee portal are higher than a business-to-consumer portal.

◆ ◆ ◆ CHAPTER 6

Implementation of a Deployment Design

During the implementation phase of the solution life cycle you work from specifications and plans created during deployment design to build and test the deployment architecture, ultimately rolling out the deployment into production.

Implementation is beyond the scope of this guide, however this chapter describes how to monitor and tune a deployment prototype.

This chapter contains the following sections:

- "About Implementing Deployment Designs" on page 99
- "Documenting the Portal" on page 100
- "Identity and Directory Structure Design" on page 103
- "Software Deployment" on page 106
- "Content and Design Implementation" on page 108

About Implementing Deployment Designs

After the deployment architecture has been approved and implementation specifications and plans have been completed, you enter the implementation phase of the solution life cycle. Implementation is a complex set of processes and procedures that requires careful planning to ensure success. Implementation includes the following tasks:

- Building the network and hardware infrastructure
- Installing and configuring software according to an installation plan
- Migrating data from existing applications to the current solution
- Implementing a user management plan
- Designing and deploying pilots or prototypes in a test environment according to a test plan
- Designing and running functional tests and stress tests according to a test plan
- Rolling out the solution from a test environment to a production environment according to a rollout plan

- Training administrators and users of the deployment according to a training plan

Details of implementation are beyond the scope of this guide. However, the following sections provide overview information for some of these tasks.

Documenting the Portal

A comprehensive set of documentation on how your portal functions is an important mechanism to increasing the supportability of the system. The different areas that need to be documented to create a supportable solution include:

- System architecture
- Software installation and configuration
- Operational procedures, also known as a "run book"
- Software customizations
- Custom code
- Third-party products integration

The run book outlines troubleshooting techniques as well as the deployment life cycle. Make this book available during the training and transfer of knowledge phase of the project.

Tip – Do not wait until the end of the deployment project, when time and money are usually running short, to begin this documentation phase. Documenting your portal should occur as an ongoing activity throughout the entire deployment.

Developing a Portal Prototype

▼ To Develop a Portal Prototype

1. **Identify and remove obvious bottlenecks in the processor, memory, network, and disk.**

2. **Setup a controlled environment to minimize the margin of error (defined as less than ten percent variation between identical runs).**

   By knowing the starting data measurement baseline, you can measure the differences in data performance between sample gathering runs. Be sure measurements are taken over an adequate period of time and that you are able to capture and evaluate the results of these tests.

   Plan to have a dedicated machine for generating load simulation which is separate from the Portal Server machine. A dedicated machine helps you to uncover the origin of performance problems.

3. **Define a baseline performance for your deployment, before you add in the full complexity of the project.**

4 **Using this initial benchmark, define the transaction volume your organization is committed to supporting in the short term and in the long run.**

Determine whether your current physical infrastructure is capable of supporting the transaction volume requirement you have defined.

Identify services that are the first to max out as you increase the activity to the portal. This indicates the amount of headroom you have as well as identify where to expend your energies.

5 **Develop and refine the prototype workload that closely simulates the anticipated production environment agreed between you and the portal administrators and portal developers.**

6 **Measure and monitor your traffic regularly to verify your prototype.**

Track CPU utilization over time. Load usually comes in spikes and keeping ahead of spikes involves a careful assessment of availability capabilities.

Most organizations find that portal sites are "sticky" in nature. This means that site usage grows over time, even when the size of the user community is fixed, as users become more comfortable with the site. When the size of the user community also grows over time a successful portal site can see a substantial growth in the CPU requirements over a short period of time.

When monitoring a portal server's CPU utilization, determine the average page latency during peak load and how that differs from the average latency.

Expect peak loads to be four to eight times higher than the average load, but over short periods of time.

7 **Use the model for long-range scenario planning. The prototype can help you understand how dramatically you need to change your deployment to meet your overall growth projections for upcoming years.**

8 **Keep the error logging level to `ERROR` and not `MESSAGE`. The `MESSAGE` error level is verbose and can cause the file system to quickly run out of disk space. The `ERROR` level logs all error conditions and exceptions.**

9 **Monitor customized portal applications such as portlets.**

10 **Monitor the following areas:**

- Sun Java™ System Access Manager
- Portal Desktop
- Sun Java System Directory Server
- Sun Java System Virtual Machine

The following sections explain issues in terms of portal performance variables and provides guidelines for determining portal efficiency.

Access Manager Cache and Sessions

The performance of a portal system is affected to a large extent by the cache hit ratio of the Access Manager cache. This cache is highly tunable, but a trade-off exists between memory used by this cache and the available memory in the rest of the heap.

You can enable the `amSSO` and `amSDKStats` logs to monitor the number of active sessions on the server and the efficiency of the Directory Server cache. These logs are located by default in the `/var/opt/SUNWam/debug` directory. Use the `com.iplanet.am.stats.interval` parameter to set the logging interval. Do not use a value less than five (5) seconds. Values of 30 to 60 seconds give good output without impacting performance.

The `com.iplanet.services.stats.directory` parameter specifies the log location, whether to a file or to the Portal Server administration console, and also is used to turn off the logs. You must restart the server for changes to take effect. Logs are not created until the system detects activity.

Note – Multiple web container instances write logs to the same file.

The cache hit ratio displayed in the `amSDKStats` file gives both an internal value and an overall value since the server was started. Once a user logs in, the user's session information remains in cache indefinitely or until the cache is filled up. When the cache is full, oldest entries are removed first. If the server has not needed to remove a user's entry, it might be the case that on a subsequent login—days later, for example—the user's information is retrieved from the cache. Much better performance occurs with high hit ratios. A hit ratio of a minimum of 80 percent is a good target although (if possible) an even higher ratio is desired.

Thread Usage

Use the web container tools to monitor the number of threads being used to service requests. In general, the number of threads actually used is generally lower than many estimates, especially in production sites where CPU utilization usually is far less than 100 percent.

Portal Usage Information

Portal Server does not include a built-in reporting mechanism to monitor portal usage information by portal users. This includes which channels are accessed, how long the channels are accessed, and the ability to build a user behavioral pattern of the portal. However, you can build a Java servlet that would intercept every Portal Server Desktop request, extract the SSO token, save the user access information to a log, then redirect the user to the intended URL. Such a construct would be based on custom attribute extensions to the Access Manager schema.

Identity and Directory Structure Design

A major part of implementing your portal involves designing your directory information tree (DIT). The DIT organizes your users, organizations, suborganizations into a logical or hierarchical structure that enables you to efficiently administer and assign appropriate access to users.

The top of the organization tree in Access Manager is called `dc=`*fully-qualified-domain-name* by default, but can be changed or specified at install time. Additional organizations can be created after installation to manage separate enterprises. All created organizations fall beneath the top-level organization. Within these suborganizations other suborganizations can be nested. The depth of the nested structure is not limited.

Note – The top of the tree does not have to be called `dc`. Your organization can change this to fit its needs. However, when a tree is organized with a generic top, for example, `dc`, then organizations within the tree can share roles.

Roles are a grouping mechanism designed to be more efficient and easier to use for applications. Each role has members, or entries that possess the role. As with groups, you can specify role members either explicitly or dynamically.

The roles mechanism automatically generates the `nsRole` attribute containing the distinguished name (DN) of all role definitions in which the entry is a member. Each role contains a privilege or set of privileges that can be granted to a user or users. Multiple roles can be assigned to a single user.

The privileges for a role are defined in Access Control Instructions (ACIs). Portal Server includes several predefined roles. The Portal Server administration console enables you to edit a role's ACI to assign access privileges within the Directory Information Tree. Built-in examples include `SuperAdmin Role` and `TopLevelHelpDeskAdmin` roles. You can create other roles that can be shared across organizations.

Creating a Custom Access Manager Service

Service Management in Access Manager provides a mechanism for you to define, integrate, and manage groups of attributes as an Access Manager service. Readying a service for management involves:

- Creating an XML service file
- Configuring an LDIF file with any new object classes and importing both the XML service file and the new LDIF schema into Directory Service
- Registering multiple services to organizations or sub-organizations using the Portal Server administration console
- Managing and customizing the attributes (once registered) on a per organization basis

See the *Sun Java System Portal Server 6 Secure Remote Access 2005Q4 Administration Guide*, *Sun Java System Directory Server Enterprise Edition 6 2006Q1 Deployment Planning Guide* and the *Access Manager Deployment Guide* for more information on planning your Access Manager and Directory Server structure.

Portal Desktop Design

The performance of Portal Server largely depends upon how fast individual channels perform. In addition, the user experience of the portal is based upon the speed with which the Portal Desktop is displayed. The Portal Desktop can only load as fast as the slowest displayed channel. For example, consider a Portal Desktop composed of ten channels. If nine channels are rendered in one millisecond but the tenth takes three seconds, the Portal Desktop does not appear until that tenth channel is processed by the portal. By making sure that each channel can process a request in the shortest possible time, you provide a better performing Portal Desktop.

Choosing and Implementing the Correct Aggregration Strategy

The options for implementing portal channels for speed and scalability include:

- Keeping processing functions on back-end systems and application servers, not on the portal server. The portal server needs to optimize getting requests from the user. Push as much business logic processing to the back-end systems. Whenever possible, use the portal to deliver customized content to the users, not to process it.
- Ensuring that the back-end systems are highly scalable and performing. The Portal Desktop only responds as fast as the servers from which it obtains information (to be displayed in the channels).
- Understanding where data is stored when designing providers, how the portal gets that data, how the provider gets that data, and the type of data. For example, is the data dynamic that pertains to an individual user, or is there code needed to retrieve that customized or personalized data? Or, is the data static and shared by a small group of users? Next, you need to understand where the data resides (for example, in an XML file, database and flat file), and how frequently the data is updated. Finally, you need to understand how the business logic is applied for processing the data, so that the provider can deliver a personalized channel to the user.

Working with Providers

Consider the following when planning to deploy providers:

- **URLScraperProvider**. Typically you use this provider to access dynamic content that is supplied by another web container’s web-based system. It uses HTTP and HTTPS calls to retrieve the content. This provider puts high requirements on the back-end system, as the back-end system has to be highly scalable and available. Performance needs to be in double-digit milliseconds or hundredths of milliseconds to show high performance. This provider is very useful for proof of concept in the trial phase of your portal deployment due to the simplicity of configuration.

`URLScraperProvider` also performs some level of rewriting every time it retrieves a page. For example, if a channel retrieves a news page that contains a picture that is hosted on another web site, for the portal to be able to display that picture, the URL of that picture needs to be rewritten. The portal does not host that picture, so `URLScraperProvider` needs to rewrite that picture to present it to portal users.

The URL Scraper provider that is part of Portal Server can also function as a file scraper provider.

To use URLScraperProvider as a file scraper provider, specify the URL as follows:

`String name="url"value="file://`*path/filename*"

This is the best performing provider, in terms of how fast it retrieves content. On the first fetch of content, performance for this provider is usually in the low teen milliseconds. On subsequent requests, using a built-in caching mechanism, this provider can usually deliver content in one millisecond or less. If applicable, consider using the file scraper provider in place of the URL Scraper provider.

- **JSPProvider**. Uses JavaServer Pages™ (JSP) technology. JSPProvider obtains content from one or more JSP files. A JSP file can be a static document (HTML only) or a standard JSP file with HTML and Java programming language. A JSP file can include other JSP files. However, only the topmost JSP file can be configured through the display profile. The topmost JSP files are defined through the `contentPage`, `editPage`, and `processPage` properties.
- **LoginProvider**. Provides access to the Access Manager authentication service through a Portal Desktop channel. This provider enables anonymous Portal Desktop login so that a user can log in directly from the Portal Desktop.
- **XMLProvider**. Transforms an XML document into HTML using an XSLT (XML Style Sheet Language) file. You must create the appropriate XSLT file to match the XML document type. `XMLProvider` is an extension of `URLScraperProvider`. This provider uses the JAXP 1.2 JAR files provided by Web Server.
- **LDAP-based provider**. This type of provider retrieves information about a user and use of personalization from user profile. It stays efficient as long as the number of LDAP attributes stored is low. In general, this type of provider is a good performer, second only to the file scraper provider within `URLScraperProvider`.
- **Database provider**. This type of provider utilizes a back-end database for its content. It requires that you build database connection polling and that you use small queries (either single queries, or no more than a couple). You might also have to perform extra work for HTML formatting. In general, this type of provider is the worst performer, due to its use of database connection pooling, large database queries, poor coding, or lack of indexing on the retrieved data. Additionally, once the data has been retrieved, the portal needs to perform a large amount of processing to display the data in the Portal Desktop. If you use this type of provider, push as much data processing logic to the database as possible. Also, benchmark your portal performance with and without database channels in the user profile.

Software Deployment

This section provides information on the software packaging mechanism, the software categories within the system, and compatibility with Java software.

Software Packaging

Portal Server uses a "dynamic WAR file" approach to deploy software to the system. Portal Server is installed using Solaris™ packages, which consist of individual files that comprise web applications, for example, JAR, JSP, template, and HTML files. The packages do not contain WAR or EAR files. The packages do contain `web.xml` fragments that are used to construct the Portal Server WAR file at installation time. This dynamically constructed file is then deployed to the web application container. As additional packages are added to the system, for example, for localization, the web application file is rebuilt and redeployed.

Note – The WAR file packaging and deployment mechanism is for use only by Portal Server products. Customer modifications to the WAR file or any files used to build it are currently not supported.

Software Categories

Portal Server distinguishes between the following kinds of software that it installs onto the Portal Server node:

- **Dynamic web applications**. These include servlets running on a Java platform, JSP files, content providers, and other items that the web container processes when accessed by the user's browser. For Portal Server, these files are installed in the Web Server.
- **Static web content**. These include static HTML files, images, applet JAR files, and other items that can be served up directly by the web server without using the Web Server container. For Portal Server, these files are also installed in the Web Server.

Note – Static web content and dynamic web applications are all grouped together into a single WAR file.

- **Configuration data**. These include data that is installed into the directory, that is, the Access Manager service definitions and any other data that modifies the directory at installation time. This includes modifications to the console configuration data to connect in the Portal Server extensions. Configuration data is installed only once no matter how many Portal Server nodes there are.
- **Software Development Kit (SDK)**. This is the JAR file or files that contain the Java APIs that are made available by a component. Developers need to install this package on a development system so that they can compile classes that use the API. If a component does not export any public Java APIs, it would not have this package.

Compatibility With Java Software

Portal Server software falls into three categories:

- Applets. Applets used in Portal Server are compatible with Java 1.1, which is supported by most browsers.
- Web applications. Web applications are intended to be compatible with the (J2EE™) web container based on the servlets interface except where uses of special interfaces are identified. This includes compatibility with Java 2 and later.
- Stand-alone Java processes. Stand-alone Java software processes are compatible with Java 2 and later. Some Portal Server software, specifically in SRA, use Java Native Interface (JNI) to call C application programming interfaces (APIs). These calls are necessary to enable the system to run as the user `nobody`.

Client Support

Portal Server supports the following browsers as clients:

- Internet Explorer 5.5 and 6.0
- Netscape™ Communicator 4.7x or higher

Multiple client types, whether based on HTML, WML, or other protocols, can access Access Manager and hence Portal Server. For this functionality to work, Access Manager uses the Client Detection service (client detection API) to detect the client type that is accessing the portal. The client type is then used to select the portal template and JSP files and the character encoding that is used for output.

Note – Currently, Access Manager defines client data only for supported HTML client browsers, including Internet Explorer and Netscape Communicator. See the Access Manager documentation for more information.

Sun Java System Portal Server Mobile Access software extends the services and capabilities of the Portal Server platform to mobile devices and provides a framework for voice access. The software enables portal site users to obtain the same content that they access using HTML browsers.

Mobile Access software supports mobile markup languages, including xHTML, cHTML, HDML, HTML, and WML. It can support any mobile device that is connected to a wireless network through a LAN or WAN using either the HTTP or HTTPS protocol. In fact, the Portal Server Mobile Access software could support any number of devices, including automobiles, set-top boxes, PDAs, cellular phones, and voice.

Content and Design Implementation

The Portal Desktop provides the primary end-user interface for Portal Server and a mechanism for extensible content aggregation through the Provider Application Programming Interface (PAPI). The Portal Desktop includes a variety of providers that enable container hierarchy and the basic building blocks for building some types of channels. For storing content provider and channel data, the Portal Desktop implements a display profile data storage mechanism on top of an Access Manager service.

The various techniques you can use for content aggregation include:

- Creating channels using building block providers
- Creating channels using `JSPProvider>`
- Creating channels using Portal Server tag libraries
- Creating channels using custom building block providers
- Organizing content using container channels

See the *Sun Java System Portal Server 6 Secure Remote Access 2005Q4 Administration Guide* and *Sun Java System Portal Server 7 Developer Sample Guide* for more information.

Placement of Static Portal Content

Place your static portal content in the *web-container-install-root*`/SUNWam/public_html` directory or in a subdirectory under the *web-container-install-root*`/SUNWam/public_html` directory (the document root for the web container). Do not place your content in the *web-container-install-root*`/SUNWportal/web-apps/https-`*server*`/portal/` directory, as this is a private directory. Any content here is subject to deletion when the Portal Server web application is redeployed during a patch or other update.

Integrating Applications

Integrating and deploying applications with Portal Server is one of your most important deployment tasks. The application types include:

- **Channel**. Provides limited content options; is not a “mini-browser”.
- Portlet. Pluggable web component that processes requests and generates content within the context of a portal. In Portal Server software, a portlet is managed by the Portlet Container. Conceptually, a portlet is equivalent to a Provider.
- **Portal application**. Launched from a channel in its own browser window; the Portal Server hosts the application; created as an Access Manager service; accesses Portal and Access Manager APIs.
- **Third-party application**. Hosted separately from Portal Server, but accessed from Portal Server; URL Scraper, which calls Rewriter, rewrites web pages so that the web pages can be displayed in a channel; uses Access Manager to enable single sign-on.

Integrating Microsoft Exchange

Using the JavaMail™ API is one of the primary options for integrating Microsoft Exchange messaging server with Portal Server. The JavaMail API provides a platform independent and protocol independent framework to build Java technology-based mail and messaging applications. The JavaMail API is implemented as a Java platform optional package and is also available as part of the Java 2 Enterprise Edition.

JavaMail provides a common uniform API for managing mail. It enables service providers to provide a standard interface to their standards based or proprietary messaging systems using Java programming language. Using this API, applications can access message stores and compose and send messages.

Independent Software Vendors

Listed below are some types of independent software vendor (ISV) integrations.

- **Application user interface**. This integration uses the provider API and SRA for secure access. (SRA is not an integration type on its own.) Examples include FatWire, Interwoven, SAP, Tarantella, Documentum, Vignette, PeopleSoft, Siebel, Citrix, and YellowBrix.
- **Security products**. This integration uses the Access Manager Login API to enable portal access by using a custom authentication scheme. Examples include RSA Security.
- **Content Management**. This integration provides data access into Portal Server, enabling searches on the data. Examples include FatWire, Interwoven, and Vignette.
- **Content Syndication**. This integration provides managing and customizing information that appears on websites. Examples include YellowBrix and Pinnacor.
- **Collaboration software**. This integration enables Sun Java System InstantMessaging product to move a collaboration session from one forum to a another. Examples include WebEx, BeNotified, and Lotus.
- **Monitoring**. This integration focuses on billing, performance measurement, and diagnostics, for which you rely on log files (or Portal Server's Logging API) and traffic snooping. Examples include Mercury Interactive, Hyperion, and Informatica.
- **Portal capability augmentation**. This integration enables products to add functionality to Portal Server. Examples include Altio, Bowstreet, rule engines to add group capability, and dynamic standard Portal Desktop and provider contents (HNC).
- **Integratable portal stack**. This integration includes products that replace elements of Portal Server. Examples include Access Manager and LDAP.

Note – Portal Server cannot currently integrate another LDAP solution. Access Manager and Portal Server rely on features not found in other LDAP implementations.

The "depth" to which user interface integration occurs with Portal Server indicates how complete the integration is. Depth is a term used to describe the complementary nature of the integration, and points to such items as:

- Application availability through Portal Server
- Application availability in secure mode (using SRA, Netlet rules)
- Ability to use single sign-on

In general, the degree to which an application integrates in Portal Server can be viewed as follows:

- **Shallow integration**. This integration essentially uses the Portal Server as a launch point. The user logs in to the portal and clicks a link that starts a web application.
- **Deep integration**. The user accesses the user interface provided by the channels in Portal Server directly. That is, the integrated software works within the portal. No additional windows or applets appear.

Rolling Out a Production Deployment

Once the pilot or proof-of-concept deployment passes the test criteria, you are ready to roll out the deployment to a production environment. Typically, you roll out to a production environment in stages. A staged rollout is especially important for large deployments that affect a significant number of users.

The staged deployment can start with a small set of users and eventually expand the user base until the deployment is available to all users. A staged deployment can also start with a limited set of services and eventually phase in the remaining services. Staging services in phases can help isolate, identify, and resolve problems a service might encounter in a production environment.

CHAPTER 7

Installed Product Layout

This appendix describes the Sun Java™ System Portal Server directory structure and properties files used to store configuration and operational data.

Directories Installed for Portal Server

Table 7–1 shows the platform-specific directory structures that are installed for Portal Server.

TABLE 7–1 Portal Server Directories

Description	Location
Default installation directory	*portal-server-install-root/SUNWportal*
Default installation directory for configuration information	`/etc/opt/SUNWportal`
Default installation directory for SDK	*portal-server-install-root*`/SUNWportal/sdk`
Temporary files	`/usr/tmp`
Log files	`/var/opt/SUNWam/log` `/var/opt/SUNWportal/`*instance-directory*
Search Engine logging, configuration, and data directories	`/var/opt/SUNWportal/`*instance-directory*`/`*log-directory*
Container and channel display profile	*portal-server-install-root*`/SUNWportal/samples/desktop/dp-org.xml`

TABLE 7–1 Portal Server Directories *(Continued)*

Description	Location
Provider display profiles	*portal-server-install-root*`/SUNWportal/samples/desktop/dp-providers.xml`
HTML template files	`/etc/opt/SUNWportal/desktop/default/`*channelname*`.template`
JSP template files	`/etc/opt/SUNWportal/desktop/default/`*JSPchannelname*
Command-line utilities	*portal-server-install-root*`/SUNWportal/bin/`
Tag library definitions	`/etc/opt/SUNWportal/desktop/default/tld/*.tld`
Display profile DTD	*portal-server-install-root*`/SUNWportal/dtd/psdp.dtd`
Java properties files	*portal-server-install-root*`/SUNWam/locale`

Directories Installed for SRA

This section describes the Sun Java System Secure Remote Access (SRA) directory structure and configuration files used to store configuration and operational data.

Table 7–2 shows the platform-specific directory structures that are installed for Secure Remote Access.

TABLE 7–2 Portal Server SRA Directories

Description	Location
Default installation directory	*portal-server-install-root/*
Default installation directory for Access Manager executable files, for the web server, and for the deployed applications	*portal-server-install-root*`/SUNWam`
Default installation directory for configuration information	`/etc/opt/SUNWportal`
Log files	`/var/opt/SUNWam/logs`

Configuration Files

All Portal Server and SRA configuration data is stored using the Access Manager Services Management function. Access Manager provides the bootstrap configuration file that is needed to find the DirectoryServer.

The `platform.conf` file contains the details that the Gateway needs. By default, the `platform.conf` file is located in `/etc/opt/SUNWportal`

Comparison of Solaris and Linux Path Names

TABLE 7–3 Comparison of Solaris and Linux Path Names

Solaris Path Name	Linux Path Name
`/opt/SUNWportal` (default)	`/opt/sun/portal` (default)
`/etc/opt/SUNWportal` (config)	`/etc/opt/sun/portal` (config)
`/var/opt/SUNWportal` (data)	`/var/opt/sun/portal` (data)

CHAPTER 8

Analysis Tools

The SunJava Enterprise System and SDK include default setting options to ensure a satisfactory out-of-the-box experience. However these options might not provide optimal performance for your web applications in the Portal Server production environment. This section describes some alternative options and basic tuning techniques.

Basic Tuning Techniques

Note – The tuning settings discussed in this section focus on Portal Server residing on the Solaris platform. However, the principles can be applied to other generic Unix type operating systems.

Table 8–1 below lists the performance analysis tools that will help in providing feedback for tuning the Portal Server and its web container. In addition to performance issues, many of these tools can be used to detect other types of bottlenecks at the overall operating system level.

Many tool descriptions provide sample output, suggestions for interpreting output results, tips on improving output results, and links to related sites.

TABLE 8–1 Performance Analysis Tools

Category	Type	Name	Parameters	Usage
Analysis Tool	Solaris 8, Solaris 9, Solaris 10	mpstat		CPU utilization
		iostat		Disk I/O subsystem
		netstat		Network subsystem
			-I hme) 10	Interface bandwidth

TABLE 8–1 Performance Analysis Tools *(Continued)*

Category	Type	Name	Parameters	Usage
			-sP tcp	TCP kernel module
			-a|grep hostname|wc-1	Socket connection count
	Portal Server on Application Server container	verbose:gc		Garbage collection
Tuning Parameters	Solaris 8, Solaris 9, Solaris 10	`/etc/system`	Various	Performance
		`/etc/rc2.d/tuning` parameters file	Various	TCP kernel tuning parameters

mpstat

The mpstat utility is a useful tool to monitor CPU utilization, especially with multithreaded applications running on multiprocessor machines, which is a typical configuration for enterprise solutions.

Use `mpstat` with an argument between 5 seconds to 10 seconds.

An interval that is smaller than 5 or 10 seconds might be more difficult to analyze. A larger interval might provide a means of smoothing the data by removing spikes that could mislead the result.

Output

```
#mpstat 10

CPU minf mjf xcal  intr ithr  csw icsw migr smtx  srw syscl  usr sys  wt idl
 0    1   0 5529   442  302  419  166   12  196    0   775   95   5   0   0
 1    1   0  220   237  100  383  161   41   95    0   450   96   4   0   0
 4    0   0   27   192  100  178   94   38   44    0   100   99   1   0   0
```

What to Look For

Note the much higher `intr` and `ithr` values for certain CPUs. Solaris will select some CPUs to handle the system interrupts. The CPUs and the number that are chosen depend on the I/O devices attached to the system, the physical location of the devices, and whether interrupts have been disabled on a CPU (`psradmin` command).

- `intr` - interrupts
- `intr` - thread interrupts (not including the clock interrupts)

- csw - Voluntary Context switches. When this number slowly increases, and the application is not IO bound, it may indicate a mutex contention.
- icsw - Involuntary Context switches. When this number increases past 500, the system is under a heavy load.
- smtx - if smtx increases sharply. An increase from 50 to 500 is a sign of a system resource bottleneck (ex., network or disk).
- Usr, sys and idl - Together, all three columns represent CPU saturation. A well-tuned application under full load (0% idle) should be within 80% to 90% usr, and 20% to 10% sys times, respectively. A smaller percentage value for sys reflects more time for user code and less preemption, which result in greater throughput for Portal application.

Considerations

Make your application available to as many CPUs as it can efficiently use. As an example, you get the best performance from one instance from 2 CPUs. You can expect that creating 14 2CPU processor sets would yield the best performance.

An increasing csw value shows an increase with network use. A common cause for a high csw value is the result of having created too many socket connections--either by not pooling connections or by handling new connections inefficiently. If this is the case you would also see a high TCP connection count when executing netstat -a | wc–l. For more information, refer to "netstat" on page 118.

Do you observe increasing icsw? A common cause of this is preemption, most likely because of an end of time slice on the CPU.

iostat

The iostat tool gives statistics on the disk I/O subsystem. The iostat command has many options. More information can be found in the man pages. The following typical options provide information on locating I/O bottlenecks.

Output

```
#iostat -xn 10
                  extended device statistics
    r/s    w/s   kr/s   kw/s wait actv wsvc_t asvc_t  %w  %b device
    0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 fd0
    2.7   58.2   14.6 2507.0  0.0  1.4    0.0   23.0   0  52 d0
   47.3    0.0 2465.6    0.0  0.0  0.4    0.0    8.8   0  30 d1
```

What to Look For

- %b - Percentage of time the disk is busy (transactions in progress). Average %b values over 25 could be a bottleneck.

- %w - Percentage of time transactions are waiting for service (queue non-empty).
- asvc_t - Reports on average response time of active transactions, in milliseconds. This option is mislabeled asvc_t; it indicates the time between a user process issuing a read and the read completing. Consistent values over 30ms could indicate a bottleneck.

Considerations

Add more disks to the file system. When using a single disk file system, consider, upgrading to a hardware or software RAID is the next logical step. Hardware RAID is significantly faster than software RAID and is highly recommended. A software RAID solution would add additional CPU load to the system.

Depending on storage hardware and application behavior, there may be a better block size to use besides the ufs default of 8192k. For more information, consult the *Solaris System Administration Guide*.

netstat

The netstat tool gives statistics on the network subsystem. It can be used to analyze many aspects of the network subsystem, two of which are the TCP/IP kernel module and the interface bandwidth. An overview of both uses follow.

netstat -I hme0 10

These netstat options are used to analyze interface bandwidth. The upper bound (max) of the current throughput can be calculated from the output. The upper bound is reported because the netstat output reports the metric of packets, which do not necessarily have to be their maximum size. The upper bound of the bandwidth can be calculated using the following equation:

Bandwidth Used = (Total number of Packets) / (Polling Interval (10))) * MTU (1500 default).

The current MTU for an interface can be found with: ifconfig -a

```
netstat -I hme0 10 Output
#netstat -I hme0 10
 input   hme0      output           input  (Total)    output
packets errs  packets errs  colls  packets errs  packets errs  colls
122004816 272   159722061 0     0      348585818 2582  440541305 2     2
0       0     0       0     0      84144   0     107695  0     0
0       0     0       0     0      96144   0     123734  0     0
0       0     0       0     0      89373   0     114906  0     0
0       0     0       0     0      84568   0     108759  0     0
0       0     0       0     0      84720   0     108800  0     0
```

What to Look For

- colls - collisions. If your network is not switched, then a low level of collisions is expected. As the network becomes increasingly saturated, collision will increase and eventually will become a bottleneck. The best solution for collisions is a switched network.
- errs - errors. The presence of errors could indicate device errors. If your network is switched, errors indicate that you are nearly consuming the bandwidth capacity of your network. The solution to this problem is to give the system more bandwidth, which can be achieved through more network interfaces or a network bandwidth upgrade. This is highly dependent on your particular network architecture.

Considerations

- If network saturation is occuring quickly (saturation at less than 8CPUs for an application server running on a 100mbit Ethernet), then an investigation to ensure conservative network usage is a good first step.
- Increase network bandwidth. Steps that possibly can be taken: upgrade to a switched network, more network interfaces are a possible solution or upgrade to a higher bandwidth network to accommodate your network traffic demand.

`netstat –sP` tcp options are used to analyze the TCP kernel module. Many of the fields reported represent fields in the kernel module that indicate bottlenecks. These bottlenecks can be addressed using the `ndd` command and the tuning parameters.

netstat -sP tcp Output

```
#netstat -sP tcp

TCP     tcpRtoAlgorithm     =      4    tcpRtoMin           =   400

        <snip>

        tcpInDupSegs        =   1144    tcpInDupBytes       =132520
        tcpInPartDupSegs    =      1    tcpInPartDupBytes   =   416
        tcpInPastWinSegs    =      0    tcpInPastWinBytes   =     0
        tcpInWinProbe       =     46    tcpInWinUpdate      =    48
        tcpInClosed         =    251    tcpRttNoUpdate      =   344
        tcpRttUpdate        =1105386    tcpTimRetrans       =   989
        tcpTimRetransDrop   =      5    tcpTimKeepalive     =   818
        tcpTimKeepaliveProbe=    183    tcpTimKeepaliveDrop =     0
        tcpListenDrop       =      0    tcpListenDropQ0     =     0
        tcpHalfOpenDrop     =      0    tcpOutSackRetrans   =    56
```

What to look for

- `tcpListenDrop` - If after several looks at the command output the tcpListenDrop continues to increase, it could indicate a problem with queue size.

Considerations:

- A possible cause of increasing tcpListenDrop is the application throughput being bottlenecked by the number of executing threads. At this point increasing application threads may be a good thing to try.
- Increase queue size. Increase the request queue sizes using `ndd`. More information on other `ndd` commands referenced in the Solaris Administration Guide.

  `ondd -set /dev/tcp tcp_conn_req_max_q` *value*

  `ondd -set /dev/tcp tcp_conn_req_max_q0` *value*

  `netstat -a | grep` *your_hostname* | wc -l

  Running this command gives a rough count of socket connections on the system. The number of connections open at one time is limited; you can use this tool to look for bottlenecks.

  ```
  netstat -a | grep <your_hostname> | wc -l Output
  ```

  ```
  #netstat -a | wc -l
  ```

  ```
  34567
  ```

What to Look For

- `socket count` - If the number returned is greater than 20,000 then the number of socket connections could be a possible bottleneck.

Consider the following:

- Decrease the point where number of anonymous socket connections start.

  `ondd -set /dev/tcp tcp_smallest_anon_port` *value*
- Decrease the time a TCP connection stays in TIME_WAIT.

  `ondd -set /dev/tcp tcp_time_wait_interval` *value*

Tuning Parameters for `/etc/system`

Table 8–2 is a list of `/etc/system` tuning parameters used during the performance study. The changes are applied by appending each to the `/etc/system` file.

TABLE 8–2 /etc/system Options

/etc/system Option	Description
set rlim_fd_max=*value*	"Hard" limit on file descriptors that a single process might have open. To override this limit requires superuser privilege.

TABLE 8–2 /etc/system Options (Continued)

/etc/system Option	Description
set tcp:tcp_conn_hash_size=*value*	Controls the hash table size in the TCP module for all TCP connections. Along with tune_t_flushr, autoup controls the amount of memory examined for dirty pages in each invocation and frequency of file system sync operations.
set autoup=*value*	The value of autoup is also used to control whether a buffer is written out from the free list. Buffers marked with the B_DELWRI flag (file content pages that have changed) are written out whenever the buffer has been on the list for longer than autoup seconds. Increasing the value of autoup keeps the buffers around for a longer time in memory.
set tune_t_fsflushr=*value*	Specifies the number of seconds between fsflush invocations.
set rechoose_interval=*value*	Number of clock ticks before a process is deemed to have lost all affinity for the last CPU it ran on. After this interval expires, any CPU is considered a candidate for scheduling a thread. This parameter is relevant only for threads in the timesharing class. Real-time threads are scheduled on the first available CPU.

A description of all /etc/system parameters can be found in the *Solaris Tunable Parameters Reference Manual*.

Table 8–3 is a list of TCP kernel tuning parameters. These are known TCP tuning parameters that affect most performance on Portal Servers.

TABLE 8–3 TCP/IP Options

TCP/IP Options	Description
ndd -set /dev/tcp tcp_xmit_hiwat 65535 ndd -set /dev/tcp tcp_recv_hiwat 65535	The default send window size in bytes. The default receive window size in bytes.
ndd -set /dev/tcp tcp_cwnd_max 65535	The maximum value of TCP congestion window (cwnd) in bytes.
ndd -set /dev/tcp tcp_rexmit_interval_min 3000	The default minimum retransmission timeout (RTO) value in milliseconds. The calculated RTO for all TCP connections cannot be lower than this value.
ndd -set /dev/tcp tcp_rexmit_interval_max 10000	The default maximum retransmission timeout value (RTO) in milliseconds. The calculated RTO for all TCP connections cannot exceed this value.

TABLE 8–3 TCP/IP Options *(Continued)*

TCP/IP Options	Description
ndd -set /dev/tcp tcp_rexmit_interval_initial 3000	The default initial retransmission timeout value (RTO) in milliseconds
ndd -set /dev/tcp tcp_time_wait_interval 60000	The time in milliseconds a TCP connection stays in TIME-WAIT state. Refer to RFC 1122, 4.2.2.13 for more information.
ndd -set /dev/tcp tcp_keepalive_interval 900000	The time in milliseconds a TCP connection stays in KEEP-ALIVE state. Refer to RFC 1122, 4.2.2.13 for more information.
ndd -set /dev/tcp tcp_conn_req_max_q *value*	The default maximum number of pending TCP connections for a TCP listener waiting to be accepted by accept(SOCKET).
ndd -set /dev/tcp tcp_conn_req_max_q0 *value*	The default maximum number of incomplete (three-way handshake not yet finished) pending TCP connections for a TCP listener.
ndd -set /dev/tcp tcp_ip_abort_interval *value*	Refer to RFC 793 for more information on TCP three-way handshake.
ndd -set /dev/tcp tcp_ip_abort_interval *value*	The default total retransmission timeout value for a TCP connection in milliseconds. For a given TCP connection, if TCP has been re-transmitting for tcp_ip_abort_interval period and it has not received any acknowledgment from the other endpoint during this period, TCP closes this connection.

CHAPTER 9

Portal Server and Application Servers

This appendix provides an overview of Portal Server and its support for application servers.

This appendix contains the following sections:

- “Application Server Support in Portal Server” on page 123
- “Portal Server on an Application Server Cluster” on page 124

Application Server Support in Portal Server

The Portal Server provides support for the following application servers to be used as the web application container, in addition to the Java™ Web Server software:

- Sun Java SystemApplication Server Enterprise Edition
- BEA WebLogic Server™ Server 8.1 SP 2
- IBM WebSphere® Application Server 5.1

Note – Portal Server runs in the context of a web application container, which can be either a web server or one of the application servers mentioned above, depending on your deployment.

Running Portal Server on an application server enables you to:

- Decouple the portal platform from the application server platform, allowing you to choose the best combination of Portal Server and application server for your organization
- Call Enterprise JavaBeans™ architecture and other J2EE™ technologies that run in the application server container
- Use application server clustering, which provides scalability and high availability
- Use session failover in clustering (currently available on BEA WebLogic Server andApplication Server Enterprise Edition)

Portal Server on an Application Server Cluster

This section describes how Application Server Enterprise Edition software, BEA WebLogic Server, and IBM WebSphere Application Server manage *application server clustering*. Application server clustering is a loosely coupled group of application servers that collaborate to provide shared access to the services that each server hosts. The cluster aims to balance resource requests, high availability of resources, and failover of application logic to provide scalability. Portal Server and Access Manager are not pure web applications. Instead, these applications are composed of local files residing on a machine and three web applications: portal, amserver, and psconsole. These three web applications run in a web application container, which runs in an application server web application container.

The Java Enterprise System installs and configures the local files, configures the local application server, then deploys the three WAR files on the local web application container. The WAR files themselves are not self-contained. The WAR files depend on the local files and directories on the machine to provide their service.

An application server cluster is a logical entity that groups many application server instances, potentially hosted on different machines. Pure web applications are deployed on a cluster using application server specific deployment tools. Once deployed on the cluster, the web applications are deployed to all the server instances that the cluster is made of, and managed in a central way.

Because of Portal Server's dual nature, as a local application as well as a web application, install Portal Server on an application server using the following steps:

▼ To Install Portal Server on an Application Server

1. **Install Portal Server on all machines using the same configuration settings.**

2. **Deploy the three web applications (portal, amserver, and psconsole) to the cluster.**

   The following sections explain what it means to enable Portal Server to run on an application server cluster.

Application Server Enterprise Edition

The Sun Java System Application Server Enterprise Edition 8 provides a robust J2EE platform for the development, deployment, and management of enterprise applications. Key features include transaction management, performance, scalability, security, and integration. The Application Server supports services from Web publishing to enterprise-scale transaction processing.

The Application Server is available in the Platform and Enterprise editions. The Platform edition is free and is intended for software development and department-level production environments. Designed for mission-critical services and large-scale production environments, the Enterprise edition supports horizontal scalability and service continuity via a load balancer plug-in and cluster management. The Enterprise edition also supports session continuity via the Highly Available Database (HADB). See the following Application Server Enterprise Edition documentation for more information:

http://docs.sun.com/db/coll/1310 ()

BEA WebLogic Server

The BEA WebLogic Server product uses the following definitions:

- **Domain**. An interrelated set of WebLogic Server resources managed as a unit. A domain includes one or more WebLogic Servers, and might include WebLogic Server clusters.
- **Administration Server**. A WebLogic Server running the Administration Service. The Administration Service provides the central point of control for configuring and monitoring the entire domain. The Administration Server must be running to perform any management operation on that domain.
- **Managed Server**. In a domain with multiple WebLogic Servers, only one server is the Administration Server; the other servers are called Managed Servers. Each WebLogic Managed Server obtains its configuration at startup from the Administration Server.

See the following documentation for more information:

`http://edocs.beasys.com/wls/docs61/cluster/index.html`

You start the Administration Server with the following command:

install_dir`/config/domain_name/startWeblogic.sh`

The local server takes its configuration from the *install_dir*`/config/domain_name/config.xml` file. To start a Managed Server, use the following command:

install_dir`/config/domain_name/startManagedWebLogic.sh`
servername admin_server_url

Instead of taking its configuration from the *install_dir*`/config/domain_name/config.xml` local file, the Managed Server takes it from the Administration Server, using HTTP.

Note – The default configuration supported for installing Portal Server on BEA WebLogic Server is a single server that is also the Administration Server for the domain.

A BEA cluster is a set of managed servers in the same domain, that are declared in the WebLogic console as a cluster. When deploying a web application, you use the name of the cluster, not the name of the individual servers. After the deployment, the web application is identically deployed to all machines in the cluster.

Session failover in BEA is described in the following document:

```
http://edocs.beasys.com/wls/docs61/cluster/servlet.html#1009453
```

Using in-memory replication for HTTP session states requires the following prerequisites:

- Portal Server supports the use of WebLogic Server clusters with in-memory session replication. See the BEA documentation for instructions to set up these clusters. The *Sun Java Enterprise System 2005Q5 Installation Guide* documents the load balancer configuration for such a cluster using the HttpClusterServlet that ships with BEA. You can also set up other load balancing hardware and software documented by BEA in the same way.
- Session data must be serializable.
- Use the `setAttribute` to change the session state.

To install a BEA cluster, your BEA license for each machine participating in the cluster must be a special BEA cluster license. See the BEA documentation for the procedure to get the license and set up a BEA cluster with HttpClusterServlet.

IBM WebSphere Application Server

The IBM WebSphere Application Server product uses the following definitions:

- **Administrative domain**. The logical space in which the configurations for various objects in the WebSphere environment reside. Inside one administrative domain you start with an application server. This is the default installation.
- **Server group**. A server group is a template for creating additional, nearly identical copies of an application server configuration. (This is the equivalent of BEA's cluster.)
- **Clones**. A copy of the server group, on the same machine or on different machines. Clones are the equivalent of BEA's managed servers.

See the IBM WebSphere Application Server documentation for more information:

```
http://www-306.ibm.com/software/webservers/appserv
```

WebSphere Advanced Server provides a more robust approach to clustering because it includes a database. In Advanced Server, all servers use the database for the configuration information. You can use the WebSphere administration console, a Swing Java application, or the command-line utilities `XMLConfig` and `wscpthen` to manage the servers.

CHAPTER 10

Troubleshooting Your Portal Deployment

This appendix describes how to troubleshoot the Sun Java™ System Portal Server software and the Sun Java System Portal Server Secure Remote Access (SRA) software.

This appendix contains the following sections:

- "Troubleshooting Portal Server" on page 127
- "Troubleshooting SRA" on page 130

Troubleshooting Portal Server

This sections contains troubleshooting information for Sun Java System Portal Server.

UNIX Processes

For the portal to be functioning properly, check that the following `root`-owned processes are running. Use the `ps` command to see this output.

Sun Java System Directory Server:

```
/ns-slapd -D /usr/ldap/slapd-server -i /usr/ldap/slapd-
server/logs/pid
```

Sun Java System Access Manager:

```
identity-server-install-root/SUNWam/bin/doUnix -c 8946
```

Sun Java System Portal Server:

```
./uxwdog -d portal-server-install-root/SUNWam/servers/https-
server/config
ns-httpd -d portal-server-install-root/SUNWam/servers/https-
server/config
```

Admin Web Server (optional, but usually running):

```
./uxwdog -d web-container-install-root/SUNWam/servers/https-admserv/config
ns-httpd -d web-container-install-root/SUNWam/servers/https-admserv/config
```

Log Files

Examine the following log files for errors.

Access Manager logs are located in:

`/var/opt/SUNWam/debug`

Portal Server logs are located in:

`/var/opt/SUNWportal/logs`

Recovering the Search Database

The Search database maintains recoverable transaction logs. Thus, under normal circumstances, you do not have to do anything to recover the database. Recovery from errors and transient conditions such as a full disk are straight forward. If desired, maintain Search database archives and restore from an archive in case you lost the entire database. In this scenario, you would copy the archive to the original database to recover it.

▼ To Recover the Database

1 **Stop all processes accessing the database, including the Portal Server instance.**

2 **Use the `rdmgr -R` command to recover.**

Working with the Display Profile

If you need to troubleshoot the XML contents of your portal's display profile, extract the contents to a file for examination. At some point in the troubleshooting process, it might be useful to reload the display profile.

▼ To Extract the Display Profile

1 **Login as administrator.**

2 **Use the `psadmin` command to extract the display profile. For example:**

```
./psadmin list -u“uid=amAdmin,ou=People,o=sesta.com,o=isp”-w
password -d“o=sesta.com,o=isp”> /tmp/displayxml
```

This example puts the contents of the display profile into the `/tmp/displayxml` file.

▼ To Reload the Display Profile

1 **Login as administrator.**

2 **Use the `psadmin` command to reload the display profile. For example:**

```
./psadmin modify -u“uid=amAdmin,ou=People,o=sesta.com,o=isp”-w
password -d“o=sesta.com,o=isp”/tmp/updated_displayxml
```

This example reloads the contents of the display profile from the `/tmp/updated_displayxml` file.

High CPU Utilization for Portal Server Instance

When using the Cisco Content Services Switch, you might see a very high CPU utilization on the Portal Server instance with Web Server error file showing the following message every five seconds.

```
[20/Jan/2003:16:53:36] failure ( 5926): Error accepting connection
 -5928, oserr=130 (Connect aborted)
```

The cause of this error is a “sticky bit” setting within the Cisco Content Services Switch that is causing these errors. These load balancers periodically ping the servers (every five seconds) to verify that the servers are alive. After turning off the “sticky bit” setting, which disables the ping to the server every 5 seconds, the errors will no longer show up in the Web Server product.

Configuring an HTTP Proxy

If the Portal Server software is installed on a host that cannot directly access certain portions of the Internet or your intranet, you can receive errors. For example, when using the `SampleSimpleWebService` provider, you might see the following error when the proxy has not been configured:

```
java.net.UnknownHostException: services.xmethods.net
```

▼ To Configure Usage of an HTTP Proxy for a Portal Server Instance

1 **Change directories to the portal server install root directory containing the configuration for the instance.**

```
cd portal-server-install-root/SUNWam/servers/https-
servername/config
```

2 **Edit the server.xml file within this directory and add the following lines:**

`http.proxyHost=`*proxy-host* `http.proxyPort=`
proxy-port `http.nonProxyHosts=`*portal-host*

where *proxy-host* is the fully-qualified domain name of the proxy host, *proxy-port* is the port on which the proxy is run, and *portal-host* is the fully qualified domain name of the portal host.

Troubleshooting SRA

This section describes how to capture information that Portal Server support personnel need to troubleshoot problems in your deployment.

Debugging the Gateway

To turn debugging on or off, you set the level of debugging or set it to off. The following steps describe what to do.

▼ To Use Debugging

1 **Log in as root to the Gateway machine and edit the following files:**

- *gateway-install-root*`/SUNWam/config/AMConfig-`*instance-name*`.properties`
- `platform.conf-default` or `platform.conf-`*gateway-instance-name-provided-during-install* file

2 **Set the debug level to FINEST.**

```
com.iplanet.services.debug.level=FINEST
```

The debug levels are:

- `FINEST` - Only serious errors are logged in the debug file. Rewriter usually stops functioning when such errors occur.
- `WARNING` - All debug messages are logged.
- `OFF` - No debug messages are logged.

3 **Set the log level in the** `platform.conf-default` **or** `platform.conf-`***gateway-instance-name-provided-during-install*.**

The log levels are:

- `WARNING`
- `INFO`
- `SEVERE`
- `FINE`

- `FINEST`

4 **Restart the Gateway from a terminal window:**

`./psadmin start-sra-instance -u` *admin-user* `-p` *passwordfile* `-N default -t gateway`

SRA Log Files

Examine the following log files for errors.

Gateway:

`/var/opt/SUNWportal/debug/srapGateway_`*gateway-hostname_gateway-profile-name*

NetFile:

`/var/opt/SUNWporta/debug/srapNetFile`

Netlet:

`/var/opt/SUNWporta/debug/srapNetlet_`*gateway-hostname_gateway-profile-name*

CHAPTER 11

Portal Deployment Worksheets

This appendix provides worksheets to help with the portal deployment process.

This appendix contains the following sections:

- "Portal Assessment Worksheets" on page 133
- "Portal Design Task List" on page 137

Portal Assessment Worksheets

Use these worksheets to learn more about your organization's business needs and potential areas of concern around deploying portals.

TABLE 11–1 General Questions

Identify the business reasons why you want a portal (check and elaborate on all that apply): Reducing procurement cost Reducing the cost of sharing information with customers, suppliers, or partners Reducing the time to deploy new business services Eliminating the cost of maintaining many point solutions Expanding the reach of the customer base for your services Securing the access to your data and services Making it easier for your customers to do business with you over the Internet Reducing the cost and time for integrating business services with suppliers and partners To comply with governmental regulations Personalizing the user experience Needing to gather business intelligence on the usage of services
How many portals does your organization already have? If you have more than one, do you have a need to reduce the number? Integrate? Federate? What types are they (business-to-employee, business-to-consumer, business-to-business, ISP)? Do you have departmental portals?
What is the extent of your Web presence? How many web sites do you have?
List the top ten application services of value to you, that you would like to expose to your partners? Suppliers? Customers? Employees?
Who is the target community for your portal?

TABLE 11–2 Organizational Questions

Who are the stakeholders of this portal?
Who are the business owners (department, organization, or an individual) within your organization who would expose the content or application service that they own by using the portal?
Would an application service exposed by using the portal be made up of smaller business applications managed by an inter-departmental business process?
Who would “own” this portal (the infrastructure)?
Who would own the content?
How do you plan to recruit additional business owners within your organization to contribute their content or applications for your portal?

TABLE 11–2 Organizational Questions *(Continued)*

What project management, architect, and technical implementation resources do you have available to help develop this portal?
Who sets the policies for web site characteristics such as look and feel and presentation?

TABLE 11–3 Business Service-level Expectations Questions

Are your development projects consistent? Do you manage their risk?
How does your development team work with your test, deployment, and operations groups?
How many different platforms does your organization currently support?
How secure is your information? How consistent is the security?
Are these challenges getting better, or getting worse?
How do you plan to recruit additional business owners within your organization to contribute their content or applications for your portal?
What project management, architect, and technical implementation resources do you have available to help develop this portal?
Who sets the policies for web site characteristics such as look and feel and presentation?

TABLE 11–4 Content Management Questions

Do you have a content or document management system?
Do you have any defined workflow to manage the development and publication of content?
Do you have a taxonomy defined?
How well is your information tagged and categorized?
How is your enterprise content developed, managed, tracked, and published?
Do you have a need for syndicated content on your portal? If so, what?
What proportion of your content is dynamic versus static?

TABLE 11–5 User Management and Security Questions

How would you segment, categorize, and relate (hierarchically) your user community?
What are your current and future security policies?
Do various departments own or maintain their private view of the customer?
Do you have an enterprise directory?

TABLE 11–6 Business Intelligence Questions

Do you have a need to gather, store, analyze, and provide information for enterprise decision-making?
Do you already employ any data analysis or OLAP tools? At what level(s) do you need to collect business intelligence (enterprise-wide, division, department, project, onetime event)?

TABLE 11–7 Architecture Questions

Do you already have an existing architecture strategy? Are there organizational issues that are hindering a successful implementation of a new IT architecture? Do you have the capabilities to implement a new architecture solution? What technologies do you currently use? Do you have the staff to implement a new architecture solution?

For the top ten services that you would like deployed by using a portal, what platform and architecture do you need to support?
How do these services authenticate users and manage access control
How do you programmatically gain access to these services?
What is your current and future messaging (email) and collaboration architecture? What is your current and future enterprise directory architecture?

What technologies are used for application integration?
What is the size of the target user community?
How many concurrent users?
What is the range of portal usage?
What is the geographical distribution of your user base?
Do you currently have or have a future need for non-Web access (Wireless, Voice/IVR)
Would your customer base require internationalization of content and services?
What server platform technologies do you use?
What development environments, tools do you use? What development methodologies do you employ?

Portal Design Task List

Table 11–8 lists the major portal deployment phases and design tasks. Use this task list to help develop your portal project plan.

Though these tasks will vary depending on your organization and the scale of each deployment, the worksheet represents the most common phases and tasks encountered.

This table consists of two columns. The first column presents the major tasks. The second column presents the subtasks for each major task.

TABLE 11–8 Design Task List

Major Phases and Tasks	Subtasks
1. Project Start and Coordination	
Project Planning	▪ Perform general project management
Project Plan Review	▪ Review pre-implementation ▪ Review business requirements ▪ Review technical requirements ▪ Review architectural documents ▪ Review hardware and infrastructure
Coordinate Resources	▪ Identify skills required ▪ Identify resources ▪ Schedule resources ▪ Assemble project team members ▪ Review work plan with project team members
Define Requirements	▪ Collect business requirements ▪ Summarize requirements ▪ Confirm functional requirements ▪ Collect technical requirements ▪ Summarize technical requirements ▪ Confirm technical requirements ▪ Prepare combined requirements document ▪ Deliver requirements
2. Design	
Develop Solution Architecture	▪ Design software architecture ▪ Design server topology ▪ Document architecture

TABLE 11–8 Design Task List *(Continued)*

Major Phases and Tasks	Subtasks
Develop Portal Integration	■ Understand system integration approach ■ Define container and channel layout ■ Define content aggregation ■ Define SSO approach ■ Develop custom Netlet and authentication modules
User Interface Design	■ Prepare or modify user interface design ■ Develop or update screen specifications ■ Review and approve user interface model
Directory Design	■ Design organizations, suborganizations, roles, and users ■ Define privileges ■ Review shared data requirements ■ Establish data transfer protocols ■ Create temporary or intermediate tables ■ Test temporary or intermediate tables ■ Document design approach ■ Deliver design document ■ Obtain appropriate stakeholder and organizational consensus
3. Develop and Integrate	
Install Software for Testing and Development Environments	■ Install Sun Java System Portal Server software and optionally Secure Remote Access software (install appropriate supporting software) ■ Install application server, if needed ■ Install other software ■ Configure server software ■ Test server software components ■ Document test findings
Install Server Software for Development Environment	■ Install Portal Server and optionally Portal Server Secure Remote Access ■ Install application server, if needed ■ Install other software ■ Test server software components ■ Document test findings
Software Configuration	■ Apply specific software configuration requirements ■ Create product configuration matrix

TABLE 11–8 Design Task List *(Continued)*

Major Phases and Tasks	Subtasks
Portal Server, Application Server, and Other Software Modifications	■ Review your organization's requirements and expectations ■ Establish modifications for software ■ Establish methods for software modifications ■ Create software modification plan ■ Design software modifications ■ Establish software modification teams ■ Create modifications ■ Test modifications ■ Obtain appropriate stakeholder and organizational review and approval of modifications
LDAP Directory Setup	■ Confer with stakeholders to establish proper schema ■ Establish modifications for software ■ Establish methods for software modifications ■ Create software modification plan ■ Design software modifications ■ Establish software modification teams ■ Create schema ■ Set up LDAP ■ Receive and verify data ■ Modify mapping as required for LDAP ■ Establish data update methods ■ Test directory ■ Create client user documentation for update methods
Legacy Software Integration (such as PeopleSoft, SAP)	■ Perform integration ■ Prepare package integration test plan ■ Perform integration test ■ Produce package integration test results
Reporting	■ Establish reporting requirements for organization ■ Create reporting plan ■ Establish reporting team ■ Design reports ■ Create reports ■ Test reports ■ Review reports with customer ■ Provide information and training on report tool
Test	■ Establish test plan

TABLE 11–8 Design Task List *(Continued)*

Major Phases and Tasks	Subtasks
Plan User Acceptance Test	■ Identify user acceptance test manager ■ Develop user acceptance test strategy and procedures ■ Review strategy and procedures with customer ■ Obtain approval for strategy and procedures ■ Develop user acceptance test roles and responsibilities ■ Obtain integration test scenarios ■ Review test conditions and acceptance criteria and revise ■ Develop user acceptance test schedule ■ Prepare acceptance test log and update with scenario test assignments
Conduct User Acceptance Test	■ Execute user acceptance test ■ Identify and document user acceptance test discrepancies ■ Resolve user acceptance test discrepancies ■ Re-execute user acceptance tests and track user acceptance test progress ■ Catalog and prioritize known limitations and process improvement opportunities identified during testing ■ Review test results with quality assurance advisors, summarize and communicate results to stakeholders ■ Obtain acceptance test approval from stakeholders
Conduct Integration and System Test	■ Ensure establishment of integration test environment ■ Identify test team and assign test scenario ownership ■ Train team on integration test procedures, roles, and responsibilities ■ Review and revise integration test execution schedule, as required ■ Execute integration test ■ Identify and document integration test discrepancies ■ Resolve integration test discrepancies and document ■ Identify required modifications (such as configuration enhancements, interfaces, reports) ■ Re-execute integration tests ■ Update as required ■ Track test progress ■ Obtain test approval ■ Summarize and communicate results to stakeholders
4. Deployment Production	

TABLE 11–8 Design Task List *(Continued)*

Major Phases and Tasks	Subtasks
Confirm Approach	■ Review with stakeholders and establish implementation locations and configurations ■ Develop implementation approach ■ Repeat appropriate tasks from development hardware and software installation
Review and Update Deployment	■ Review existing documentation of results of tests ■ Validate scope, objectives, and critical success factors ■ Update deployment approach ■ Review and approve deployment
Implement Deployment	■ Review and reconcile system operations ■ Review organization and system procedures ■ Promote to production ■ Update current operations ■ Revise system release and deployment materials ■ Provide transition support
Training	■ Confirm organization commitment and expectations ■ Establish training requirements for all personnel ■ Establish training schedules ■ Establish training staff ■ Prepare materials for training ■ Train administrators ■ Train maintenance providers ■ Capture training feedback ■ Incorporate feedback for training improvement
Document Portal	■ Create a “run book” for system administrators

Index

A

accelerators, and Gateway, 55
access control
  Gateway, 55
  limiting, 47
  NetFile, 59
  Netlet, 57
Access Control Instructions, 103
Access Manager
  administration console, 51-52
  cache and sessions, 102
  components, 51-52
  customizing, 103-104
  description, 21
  description and benefits, 21
  organization tree, 103
  single sign-on, 51-52
Access Manager SDK, components, 62
administration console tasks, 51-52
aggregation
  description and benefits, 26
  strategy, 104
Allowed URLs and Denied URLs lists
  Gateway, 55
  NetFile, 59
`amSDKStats` log, 102
`amSSO` log, 102
analysis tools, 115
anonymous Desktop, 105
applets, NetFile, 58
application servers
  clustering, 124-126
  requirements, 93
  support for, 123
applications
  degree of integration, 110
  dynamic port, 56
  integrating, 108-110
  portal, 108
  static port, 56
  third-party, 108
authentication, 21, 46-47, 105
  and LDAP, 64
  basic authentication, 78
  custom, 109
  Gateway, 54
  modes, 54
  PDC, 54
  Portal Server, 46-47
  UNIX, 58
authentication server, 65
average session time, 36
average time between page requests, 35

B

back-end servers, 38
banner, 52
baseline portal performance analysis, 100
basic authentication, 54
BEA WebLogic, 125-126
bottlenecks, and building modules, 74
building modules, 67-75
  and Directory Server, 72
  and high availability, 68-73
  and Search Engine, 75
  and transparent failover, 72-73

building modules (Continued)
- deploying, 74-75
- description, 68

business-to-consumer portal, 97
business-to-employee portal, 97

C

cache hit ratio, 102
caching appliance, and reverse proxy, 52
channels
- description, 108
- organizing content, 108

checkpointing mechanisms, 69
chroot environment, 53
Citrix, 30
client detection API, 107
client support, 107
clustering
- application servers, 123, 124-126
- session failover, 123

communication links, 63
components
- Access Manager Server, 51-52
- NetFile, 58
- Portal Server, 51-52
- SRA, 52-61

concurrent sessions, 34-35, 35-36
concurrent users, 35-36
configuration data, 106
configuration files, Portal Server and SRA, 112
configuring, HTTP proxy, 129-130
content, placing, 108
CPU utilization
- and mpstat utility, 116
- high with Cisco Content Services Switch, 129

CPUs
- and Gateway instances, 40
- and vertical scaling, 45
- estimating number, 34

credentials, NetFile, 58
customizing
- Access Manager service, 103-104
- affects on performance, 93
- baseline figures, 93

D

data centers, and sizing, 94
Database provider, 105
delegated administration, 22
Demilitarized Zone, description, 51-52
deployment
- bottlenecks, 74
- building modules, 74
- ISP hosting, 66
- providers, 104-105

deployment scenarios, 69-70
- and SRA, 70
- building modules, 69
- no single point of failure, 70
- transparent failover, 72-73

designing
- for localization, 48
- security strategies, 46-48
- use case scenarios, 42-44

Desktop type, 37
directories
- installed for Portal Server, 111-112
- installed for SRA, 112

Directory Information Tree, 103
directory replica, 72
Directory Server
- and building modules, 72
- clustering, 69
- description, 51-52
- structure design, 103-104

Directory service, description, 21
directory structure, SRA, 112
Discussion channel, 24-25
display profile, 108
- and JSP files, 105
- and troubleshooting, 128-129
- DTD location, 112
- extracting, 128-129
- location for provider, 112
- properties, 48
- reloading, 129

DIT, 103
DMZ, description, 48
documenting the portal, 100
`dp-org.xml` file, 111
`dp-providers.xml` file, 112

dpadmin command, 128, 129
dynamic port applications, 56
dynamic web applications, 106

E

encryption, 46-48
  128-bit, 78
  40-bit, 55
  Netlet, 55
  Portal Server, 46-47
  symmetric key encryption, 96
Enterprise JavaBeans, 93
error logging level, 101
/etc/opt/SUNWps directory, 111
/etc/system tuning parameters, 120
example use case, 43-44
extracting the display profile, 128-129

F

failover, 45, 69
file compression, NetFile, 59
FTP, NetFile, 58

G

Gateway
  accelerators, 55
  access control, 55
  Allowed URLs and Denied URLs, 55
  and HTTP basic authentication, 54
  and Non Authenticated URL, 55
  and proxies, 65
  authentication, 54
  chroot environment, 53
  high availability, 45
  HTTP and HTTPS, 53
  logging, 55
  multihomed, 53
  multiple instances, 53
  Netlet traffic, 55
  overview, 53-55

Gateway (Continued)
  page configuration, 37
  profile, 54
  proxies, 53
  session
    information, Gateway, 54
  session stickiness, 53
  SSL, 54-55

H

hardware redundancy, 69, 70
heap size, 95
high availability
  and building modules, 68-73
  and Portal Server components, 45
high-level architecture,typical installation, 66
high-level portal design, overview, 50
horizontal scaling, description, 45-46
HTTP and HTTPS modes, and Gateway, 53
HTTP basic authentication, 54
HTTP proxy, configuring, 129-130
HttpSession failover, 69

I

IBM WebSphere Application Server, overview, 126
Identity management, features and benefits, 20-22
independent software vendors, types, 109-110
installing, as a regular user, 47
integrating applications, 108-110
interface bandwidth, and netstat, 118
Internet Explorer, 107
iostat tool, 117
ISP hosting deployment, 66
isp organization, 103
ISVs, types, 109-110

J

Java compatibility, 107
Java properties files, 112

JavaScript
- in Rewriter, 60
- Portal Server Desktop, 52

JavaServer Pages, 105
JAXP, 51-52
JCA, and Refine, 93
jCIFS, NetFile, 58
JDBC, and sizing, 93
JSP template files, location, 112
JSPProvider, 105

L

LDAP
- authentication, 64
- transaction numbers, 93

LDAP-based provider, 105
LDIF file, 103
legacy servers, 62
load balancing
- and high availbility, 69
- and Portal Server failures, 71
- and Rewriter, 61
- and SRA, 45, 72

`locale` file, 112
localization, 48
log files
- and troubleshooting, 128
- location, 111
- SRA, 131

logging
- errors, 101
- Gateway, 55
- number of active sessions, 102

login type, 37
LoginProvider, 105
low-level portal design, overview, 50

M

Microsoft Exchange, 56
- and Netlet, 56
- integrating, 109
- Netlet Proxy, 57

MIME types,NetFile, 59
monitoring, active sessions, 102
mpstat, 116-117
multi-master
- and Directory Server, 69
- configuration, 69, 74

multihomed Gateway, 53
multiple network connections, Portal Server, 57
multithreading
- and mpstat, 116
- NetFile, 60

N

NetFile
- access control, 59
- Allowed URLs or Denied URLs, 59
- applet, 58
- components, 58
- compression, 59
- compression types, 59
- initialization, 58
- multithreading, 60
- overview, 57-60
- Portal Server Desktop, 58
- search, 59
- security, 59
- validating credentials, 58

Netlet
- access control, 57
- and Microsoft Exchange, 56
- and third -party applications, 56
- application integration, 56
- encryption, 41
- encryption (ciphers), 55
- overview, 55-57
- requests and Gateway, 53
- split tunneling, 56-57
- traffic, 55
- usage characteristics, 41

Netlet Proxy
- and software crash, 45
- and transparent failover, 73
- Microsoft Exchange, 57
- overview, 57

Netlet Proxy (Continued)
- third party proxy, 57

Netscape Communicator, 107
netstat tool, 118
NFS, NetFile, 58
Non Authenticated URL list, and Gateway, 55
Novell domain, 58

O

`/opt/SUNWps` directory, 111
`/opt/SUNWps/sdk` directory, 111
Outlook client, 56

P

packaging, 106
pcAnywhere, 30
PDC authentication, 54
peak numbers, 34-35
performance
- Access Manager cache and sessions, 102
- analysis tools, 115
- building modules, 74
- garbage collection, 95
- memory consumption, 95
- TCP kernel, 121
- thread usage, 102
- tuning parameters, 120

personalization
- description and benefits, 25-26
- retrieval, 105

placement of portal content, 108
Portal Desktop
- configuration, 37
- design, 104-105

portal key design task list, 137-141
Portal Server
- and Access Manager on different nodes, 62-63
- and high availability, 45
- and load balancers, 71
- building modules, 67-75
- client support, 107
- communication links, 63

Portal Server (Continued)
- components, 51-52
- configuration files, 111
- design approach, 92
- directory structure, 111-112
- documenting functions, 100
- hardware and applications, 38
- high-level design, 50
- instance and servlets, 64
- logical architecture, 51-52
- low-level design, 50
- multiple network connections, 57
- nodes, 53, 62
- overview, 19
- scalability, 45-46
- security, 46-47
- SRA overview, 20
- stickiness, 101
- troubleshooting, 127-130
- tuning goals, 44
- typical installation, 66
- usage information, 102

Portal Server Desktop
- JavaScript, 52
- NetFile, 58

portlet, description, 108
profile database server, 65
Provider Application Programming Interface, 26
providers, deployment considerations, 104-105
proxies, 53
- and Gateway, 65
- configuration, 54
- failover, 45

Proxylet, overview, 61
`psdp.dtd` file, 112

Q

questions, user behaviors and patterns, 31

R

`rdmgr` command, 128
recovering, Search database, 128

reloading the display profile, 129
resource bundles, 48
reverse proxy
  description, 89-90
  offloading requests, 52
Rewriter
  load balancing, 61
  overview, 60-61
  rulesets, 60
Rewriter Proxy
  and accelerators, 96
  and software crash, 45
  overview, 61
robot, 24
Role-Based Access Control, 47
roles, 103
rulesets, Rewriwter, 60

S
scalability, 45-46
  and SRA, 40
  portal channels, 104
SDK, description, 106
search, NetFile, 59
Search database
  and robot, 24
  recovering, 128
Search Engine
  description and benefits, 24-25
  functions, 36
search engine, sizing factors, 36
Search Engine
  structure, 75
searchURL property, 75
security, 46-47
  NetFile, 59
security strategies, 46-48
servlets, and communication, 64
session
  characteristics, SRA, 41
  monitoring, 102
  stickiness, 53
session failover, 69
  and clustering, 123

session failover (Continued)
  BEA, 125
session information, 54
single sign-on, 46
  description, 22
sizing, 93
  and JCA, 93
  and JDBC, 93
  refining, 94
  search engine factors, 36
  tool, 40-41
  validating, 93-94
software
  categories, 106
  packaging, 106
software crash, 45
Solaris Operating Environment, minimizing size of installation, 46-48
split tunneling, 56-57
SRA
  and load balancing, 45, 72
  and NetFile, 59
  and reverse proxy, 89-90
  and Sun Enterprise Midframe Line, 96
  components, 52-61
  debugging, 130-131
  directory structure, 112
  features and benefits, 22
  log files, 131
  overview, 20
  session characteristics, 41
  troubleshooting, 130-131
SSL
  encryption, 46-48
  Gateway, 54-55
  modes, 54
  v2 and v3, 54
state data, and Portal Server services, 65
static port applications, 56
static portal content, 108
static web content, 106
subscription channel, 24-25
Sudo, 47
Sun Cluster software, 69
Sun Crypto Accelerator 1000 board, 96
Sun Java System Application Server, overview, 124-125

SuperAdmin Role, 103
system capacity, 39
system performance, 39

T
tag library definitions, 112
task list, 137-141
TCP kernel tuning parameters, 121
third-party applications
  and Netlet, 56
  description, 108
third party proxy, Netlet Proxy, 57
thread usage, 102
transaction time, 38
transparent failover, and building modules, 72-73
troubleshooting, 127-130, 131
  SRA, 130-131
tuning
  goals, 44
  settings, 115
tunneling, 56-57

U
UNIX
  authentication, 58
  user installation, 47
UNIX processes, troubleshooting, 127-128
usage information, 102
use case scenarios
  designing, 42-44
  example, 43-44
user behaviors and patterns, 31

V
vertical scaling, description, 45
VPN, 23
VPN client, 56

W
WAR file, 106
  and application servers, 124
  to deploy software, 106
web containers, supported, 123
workload conditions, 39
worksheets, 133-141

X
XMLProvider, 105