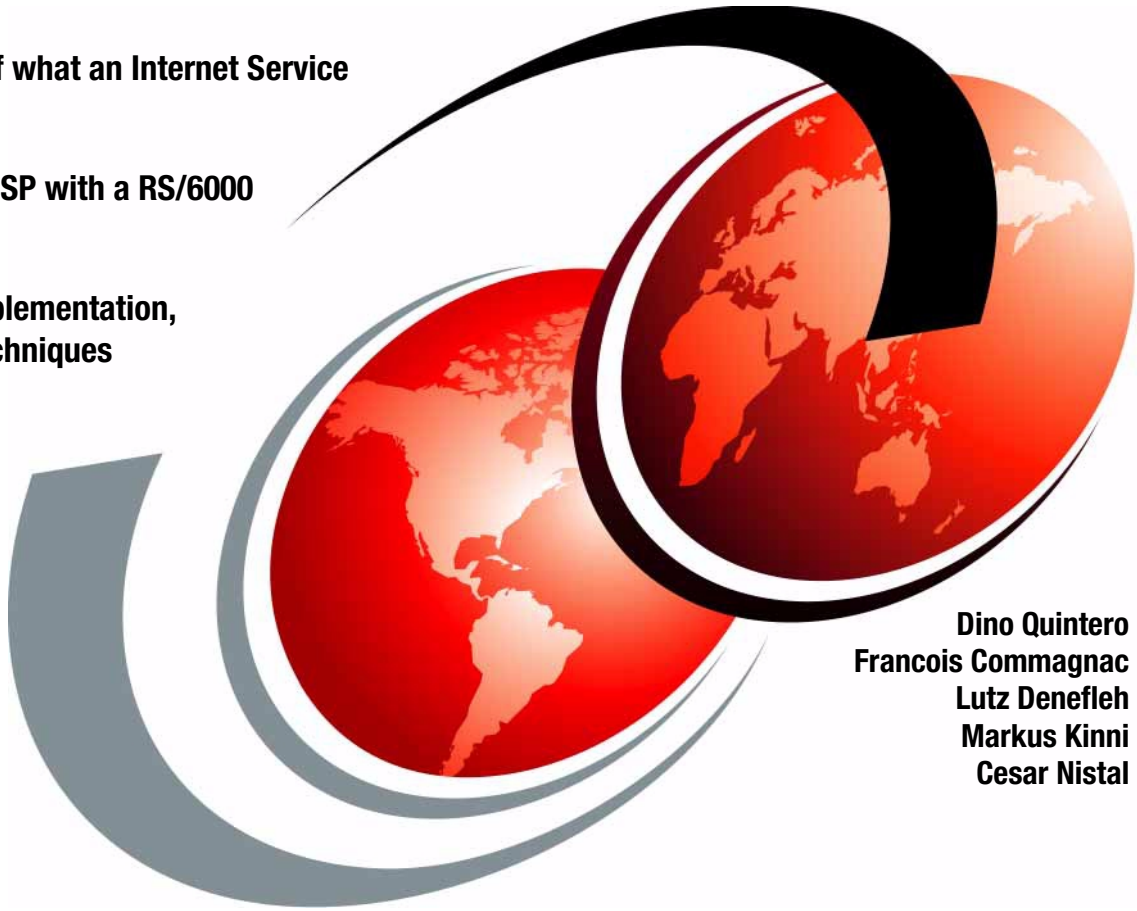
IBM

Integrating an ISP into a RS/6000 SP Environment

- Overview of what an Internet Service Provider is
- Grow your ISP with a RS/6000 SP
- Sample implementation, tips and techniques



Dino Quintero
Francois Commagnac
Lutz Denefleh
Markus Kinni
Cesar Nistal

Redbooks

ibm.com/redbooks

SG24-6025-00

International Technical Support Organization

Integrating an ISP into a RS/6000 SP Environment

January 2001

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix C, “Special notices” on page 303.

First Edition (January 2001)

This edition applies to Version 3, Release 2 of the IBM Parallel System Support Programs for AIX (PSSP) Licensed Program (product number 5765-D51), and AIX Version 4 Release 3 Licensed Program (product number 5765-C34).

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. JN9B Building 003 Internal Zip 2834
11400 Burnet Road
Austin, Texas 78758-3493

Contents

Figures

Tables

Preface

This redbook takes you partly inside the Internet Service Provider (ISP) business and environment. It delivers an overview on what is an ISP, what are the technologies in place and how they work, what is coming in the near future, and how it is affecting the ISP business.

The redbook focuses on integrating an ISP environment into an RS/6000 SP environment. It identifies and implements the necessary hardware and software that are required for setting up an ISP configuration. This redbook describes the initial sizing, configuration guidelines, and step-by-step procedures for installation, mail configuration, and news, web and directory services that are required for an ISP solution.

The introduction defines the basic rules of the ISP market. It then defines the architecture and components for an ISP platform. Three solutions sets are proposed, and possible growing paths are described.

This redbook does not replace the latest RS/6000 marketing materials and tools. It is intended as an additional source of information that, together with existing resources, may be used to enhance your knowledge of IBM solutions for the UNIX marketplace.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization Austin Center.

Dino Quintero is a project leader at the International Technical Support Organization (ITSO), Poughkeepsie Center. He has over nine years experience in the Information Technology field. He holds a BS in Computer Science and a MS degree in Computer Science from Marist College. Before joining the ITSO, he worked as a Performance Analyst for the Enterprise Systems Group, and as a Disaster Recovery Architect for IBM Global Services. He has been with IBM since 1996. His areas of expertise include enterprise backup and recovery, disaster recovery planning and implementation, and RS/6000. He is also a Microsoft Certified Systems Engineer. Currently, he focuses on RS/6000 Cluster Technology by writing redbooks and teaching IBM classes worldwide.

Francois Commagnac is a NetGen IT Architect at the e-Business Solution Center in La Gaude, France. He holds a Graduate Engineer in computer science. Through five years of working experience in areas such as Optical

Simulation, CAD, Telco and Media, he has gained extensive knowledge on middleware such as CORBA and OO designs, and has leading designs on distributed application servers using these technologies. He joined NetGen at the beginning of year 2000, has actively worked to define the ParaBlue solution for rapid ISP deployment, and has helped customers in the ISP, Mobile Portals, and ASP segments.

Lutz Werner Denefleh is a team leader at the Integrated Technology Solutions (ITS) Central Region, Mainz Support Center. He holds a Graduate Engineer in Fluid Dynamics. He has 12 years of experience in the Information Technology field. He has worked at IBM for 11 years. His areas of expertise include solution implementations on RS/6000, such as CATIA, Lotus Notes/Domino, and Tivoli. In 1994, he started to work with Internet technologies. He is now responsible for the IBM Mainz Support Internet link, which includes hosting services for several IBM Webservers using SP2.

Markus Kinni is an IT Architect at the IBM Global Services (IGS) in Finland. He has three years of experience in the Information Technology field. He holds a degree in Computing Science from Helsinki University in Finland. His areas of expertise include middleware technologies and different wireless solutions based on pervasive devices and wireless networks. He has two years experience working with radio networks and Wireless Local Area Networks (WLAN). He is currently working with the WebSphere product family, mostly with Tivoli Internet Subscriber Management (TiSM) and WebSphere Everyplace Suite (WES).

Cesar Nistal Tome is a Systems Engineer at IBM Global Services, Spain. He holds a degree in computing engineering from Oviedo University in Spain. He has over three years working in the AIX and SP fields and was involved in the Sydney Olympic Project for two years. He is a Certified Advanced Technical Expert.

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

International Technical Support Organization, Poughkeepsie Center
Subramanian Kannan, Yoshimishi Kosuge

IBM Madrid, Spain
Ivan Montesino Martinez del Cerro

IBM Poughkeepsie
Ed Merenda, Chris Hawkinson

IBM Hursley, PISC, UK
Jim Hall, David Crowther, Jon P. Harry

IBM Markham, USA
Sam Iskandar

IBM Cary, USA
Clem Leung

IBM Australia
James Kelly

IBM Philippines
Dominique Cimafranca

IBM La Gaude, France
Clemens Harald Engler, Santina Franchi, Jose Martinez, Francois-Xavier Drouet, Fabien Lanne, Denis Chevallier

ISV Center for NetGen, IBM France
Didier Kusberg, Julien Aicart, Jean-Yves Girard, Eric Larthe, Yann Guerin

IBM Austin, USA
Ron Gordon

Comments welcome

Your comments are important to us!

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

- Fax the evaluation form found in “IBM Redbooks review” on page 329 to the fax number shown on the form.
- Use the online evaluation form found at **ibm.com**/redbooks
- Send your comments in an Internet note to redbook@us.ibm.com

Chapter 1. What is an Internet Service Provider environment?

This redbook provides an overview on how to create an Internet Service Provider (ISP) environment in an RS/6000 SP environment. First, we define some of the rules of the ISP market, then we define the architecture and components for an ISP platform. Three solution sets are proposed in this redbook and possible growing paths are described.

1.1 ISP definition

The ISP definition provided at `www.webopedia.com` is:

> Short for Internet Service Provider, a company that provides access to the Internet. For a monthly fee, the service provider gives you a software package, username, password and access phone number. Equipped with a modem, you can then log on to the Internet and browse the World Wide Web and USENET, and send and receive e-mail.
>
> In addition to serving individuals, ISPs also serve large companies, providing a direct connection from the company's networks to the Internet. ISPs themselves are connected to one another through Network Access Points (NAPs).

The ISP market is evolving very quickly. The definition of an ISP is not as clear as it used to be. For example, new access methods have been implemented, such as Digital Subscriber Line (DSL) and cable access. New types of devices have access to the Internet, like mobile or screen phones. The number of basic services that are the root of the ISP business have increased due to fierce competition, new technologies and better access to more bandwidth. Even the business environment, with the liberalization of Telco companies, for example, is changing.

We define, in the following paragraphs, what we think the market looks like today. We then outline what part of the market we cover. Most importantly, we give a perspective of the requirements an ISP can or has to cover in order to succeed.

By describing our view of the ISP market today, we give the definition of an ISP, or more likely, the definitions of an ISP. There are no identical ISPs as one solution does not fit all. The ISPs described here are no exception to that rule. Each solution is a one of a kind solution.

1.2 Market

At the time of this redbook, we provide an instant market outlook of what an ISP is today. We know from experience that the market evolves in an unpredictable way. This has been the case over the past four years. Nevertheless, the market itself has changed in such a way that it is easier today to forecast what the market will become tomorrow. The basic lines of what an ISP is have been clearly defined and the new entrants on this market have been identified and understood. The technologies used in the Web environment are still quickly evolving, but the business models using them are known. The ISP model is, in this respect, an old segment, especially compared to new ones in the Business-to-Business (B2B) segment, or the always innovative Industry.com sector (see Section 1.2.4, “Segmentation” on page 4 for more details on these segments). These other models are out of the scope of this book. Our aim is to define, in clear boundaries, what is going to be the scope of this redbook in terms of added value services.

1.2.1 The global picture

Companies in all sectors of the economy are making the transition to a networked, global economy. Recognizing that web-based technologies are becoming key enablers to their mission critical business processes, many companies are moving beyond basic Internet services to more complex e-business services.

Companies seeking to use e-business to streamline and optimize their businesses face a new set of challenges, for which they are not always well prepared. These challenges include the following:

- Economics

  e-business is fundamentally altering the underlying economics of business. Web-based technologies are providing new opportunities for companies to increase their efficiency by eliminating significant cost elements from their business models while expanding the scope and scale of their operations.

- Competition

  The Web is intensifying competition. Internet-based products, services and channels increase competition by lowering barriers to entry and blurring traditional industry distinctions. In addition, as early adopters start to reap the benefits of their Internet investments, competitors are coming under intense pressure to initiate some kind of response.

- Customer Value

  Web technology has created a new, direct channel to the consumer. The penetration of the Internet into the marketplace has increased the opportunity to effectively channel marketing efforts.

1.2.2 Expansion throughout geographies

The Internet market has attracted a large number of new entrants because it is a high growth market with relatively low barriers to overcome. Although the worldwide market is highly fragmented, there is a distinct group that accounts for the vast majority of the total revenue and is very influential in the overall market. Driven by the need to gain cost advantages, these large players are aggressively seeking economies of scale and market share. As a result, the industry is beginning to experience significant consolidation.

1.2.2.1 North America market

In North America, there are approximately 4,500 small ISPs (less than 10,000 subscribers) delivering basic Internet services with an estimated turnover of 1,000 ISPs each year. Acquisition and consolidation of small ISPs continues to be a major trend. Takeover of ISP services has been highest in North America, particularly in the U.S., and is dominated by a few key continent wide players. Several emerging and established service providers are continuing to launch ISP services from their home base of North America to provide services in Europe, Asia-Pacific, and Latin America.

1.2.2.2 European market

The European Internet market has began to expand quickly and is dominated in most regions by incumbents. Free Internet access has been driving some of this growth, due to local phone access charges in most European countries. ISPs in this region are turning to value-added services to generate profit. Telco companies have emerged as major actors in this market because they have the infrastructure and experience to venture successfully on that space (as an added service of their existing portfolio).

1.2.2.3 Caribbean and Latin America (CALA) market

The CALA market is just beginning to embrace the Internet. Issues of low incomes, limited communications infrastructure, and regulated markets are being overcome, and the region is showing strong Internet subscriber growth in most countries. Free Internet access is helping to drive this, but value-added services is what will sustain new ISPs in the long term.

1.2.2.4 Asia-Pacific market

Asia-Pacific is still the sleeping giant, ripe for its own Internet boom. Many of the Asian markets are deregulated; it is a matter of time before they are also able to take advantage of the Internet on a large scale.

1.2.3 Traditional structures

Traditional ISPs are primarily involved in the provision of basic Internet services, such as dial-up access and e-mail. Consequently, they generated most of their revenue from access and subscription fees. Over the last four years, customer demand for both basic access and a wide range of value-added Internet services has expanded rapidly. As Internet access is becoming a commodity, differentiation in that market segment must come from new services. At the same time, new business models providing innovative offerings to customers are created and implemented. This generated even more demands from the consumers. The breadth of possibilities has expanded beyond any expectations. The phenomenal growth of the market, combined with relatively low barriers to overcome, has attracted an influx of new participants.

First generation ISPs have now been joined by a large number of companies. These companies have different strategic objectives and bring with them new business models and competencies. The distinction between various types of service providers is blurring. What was previously referred to as ISPs are becoming Application Service Providers (ASPs), Telcos are offering ISP services, and Network Service Providers (NSPs) are trying to “look local” by providing regional specific services to compete with smaller service providers as well as providing their traditional backbone services. Competition between service providers is driving down the price of Internet access to free access, forcing ISPs to offer additional differentiated and end-to end solutions through more valued services.

1.2.4 Segmentation

To look at the current segmentation of the ISP business activities, we need to look at the traditional segmentation of the new business economy. IBM has launched the interNET GENeration (NetGen) initiative to help old and new companies take their businesses on the Internet. We will describe the NetGen Traditional Segmentation and then we will define segmentation that is more specific for ISPs.

NetGen traditional segmentation

NetGen has proposed a segmentation approach to help identify businesses’ goals and models and the needs and the infrastructure requirements for

these companies. This is shown in Section 1.2.4, “Segmentation” on page 4. The segments are divided in two major groups:

- Service providers

  These companies deliver intelligent infrastructures

  - Telco NSP

    Subsidiary or division of Regional Bell Operating Companies (RBOCs), Internet eXChange (IXCs) and Post Telegraphy Telecoms (PTTs), or independent Telcos that provide Internet-based services to their customers, like access, hosting, messaging, commerce, local and regional portals.

  - Access ISP

    Provides end-customer access to the Internet. For consumers, this is primarily dial-up, with emerging high-speed access delivered via DSL (Telcos) or cable modems (cable companies). For businesses, this is primarily via leased line connections.

  - Wholesale ISP

    Provides services to another segment of the NetGen industry. Today these are primarily backbone network services, but there are emerging wholesale offerings for value-added services, such as hosting.

  - Web Hoster

    Provides and manages the Web sites and applications for end customers' web presence. These services are typically delivered in three distinct models (shared services, dedicated services, and co-location).

- Internet companies

  - Application Service Providers (ASPs)

    This emerging segment provides hosted applications that are managed by the ASP and delivered via contract on monthly fee, subscription, or rental basis to end customers (primarily businesses). ASPs are included in a rich variety of business models that tend to cover more and more segments.

  - Portal

    Provides navigation tools and access to content on the web. Because of their access to “eyeballs,” portals have been able to generate significant revenue from advertising. Differentiated from *Industry.com* by their broad audiences focus (consumers or business users).

- eMP

  Market places are a strongly emerging segment where professionals can exchange, auction or trade goods. eMPs are meant to enable supply chain management activities on the Internet.

- Born on the Web

  These companies build their business model with the Internet as the primary infrastructure to support most business processes. Most of these companies sell services and products. It is also difficult to categorize the business model of these companies. These are companies that are “born on the web” (start-up) or “born in the board room.”

This segmentation has evolved since it was first proposed, as shown in Figure 1. It is still changing today.

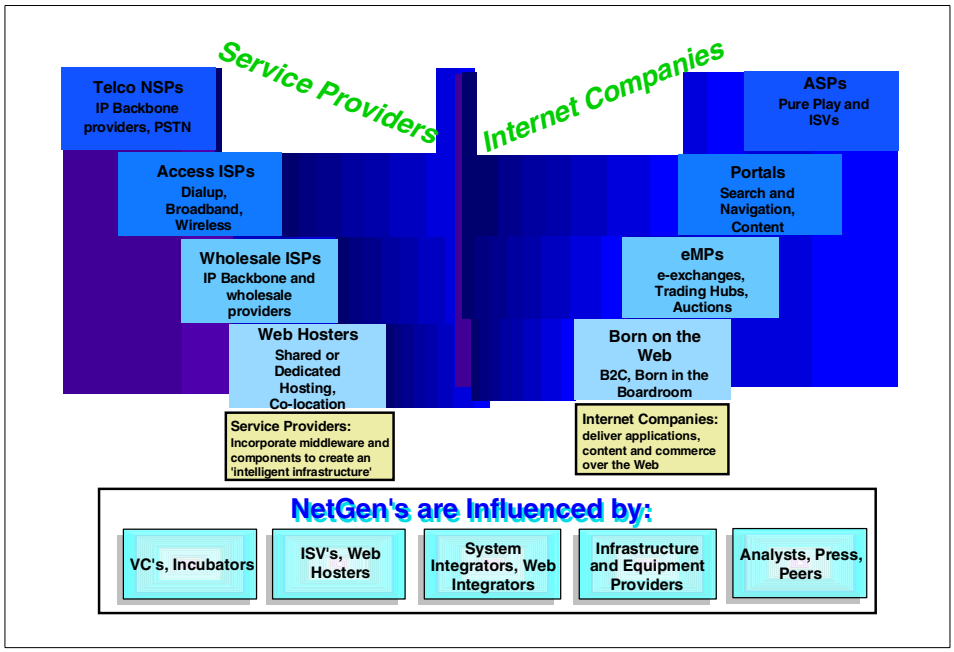


Figure 1. NetGen segmentation

ISP segmentation today

We have presented the complete list of NetGen segments because ISPs are showing more and more determination in entering other segments to conquer new market opportunities.

The first commercial providers were independent ISPs and Telcos. Today, these companies have been joined by a large number of new participants, including cable operators, media companies and content providers.

With these new actors, the barriers between the different segments are fading. For example, Telco ISPs become Access ISPs, and Access ISPs deliver ASPs capabilities.

The ISP segments are distinct in their competitive focus:

- *Transport providers* are ISPs that focus on selling wholesale bandwidth and point of presence (POP) to other ISPs and large businesses. These ISPs own large backbone network facilities, which they develop, build, maintain, and operate. They can provide direct access to their backbone to larger businesses becoming access ISPs, even though their primary business model is wholesale ISP.
- *Hosting services* (Internet Data Center (IDC) and Infrastructure operators) provide for and/or manage server farms for content and application hosting. The primary business model for these companies is Web Hosting. With some technology leveraging, they can adapt to become well-positioned ASPs due to the capability of their infrastructure. They are high performance nodes with structured practices. They can host, in time, any kind of Internet service from traditional ISPs and ASPs to Industry.coms.
- *Packagers (*application and content providers*)* are customer-focused ISPs that interface directly with consumers, providing content, application function and Internet access. These consumers can either be residential users, businesses, or both. Because of their audience, especially business consumers, they are well-positioned to do B2B activities and ASP. Larger infrastructures can also do some Web Hosting. Packagers are also Telco ISPs. Telco ISPs are even better positioned, with existing telephone and/or mobile customer bases and wide POP. The leveraging to Internet technologies is more painful because of the legacy IT infrastructure.

We will concentrate on the *packagers* model. We describe the solutions to set up an ISP and what added-value services are available to increase their revenue stream. We will provide, in Section 1.2.7, “Internet Data Centers (IDC)” on page 11, an overview on IDCs and what services they can deliver. RS/6000 SPs are well suited to host efficiently applications in this business model.

1.2.5 Packagers business model

Packager ISPs are offering a wide range of services, from basic access to complex value-added services. Although there is still strong growth in basic access, margins have declined significantly due to competitive pressures. Many Packager ISPs have begun to price this service at cost in order to “buy” market share. Packager ISPs are looking to diversify their revenue streams with higher margin, value-added services.

There are two main business models in the packagers landscape:

- Packager ISPs for residential users

  Packager ISPs gain high visibility and large audiences in this market. Large audiences enable revenues from advertising, partnerships and commerce activities.

- Packager ISPs for businesses

  Packager ISPs gains different sources of revenue through the billing of premium services with a high potential of margin. Billing is discussed in more detail in Section 2.8.2, “Billing systems” on page 98.

Both models can be combined into one, as Packager ISPs can address both residential and business users. This model combines high visibility and flat rate revenue stream from businesses.

Figure 2 on page 9 shows a model of the grid of services that can be offered by an ISP to the business consumer market. Transport Providers are, as shown, best fit to provide some of services businesses may require. Hosting services are not shown, as they do not provide direct access to the Internet for consumers; their business is to host brands or application.

Residential users offered are shown in this model. This is why both markets can be reached by an ISP through a common set of services. The services that fit most residential and business users is also depicted in Figure 2 on page 9.

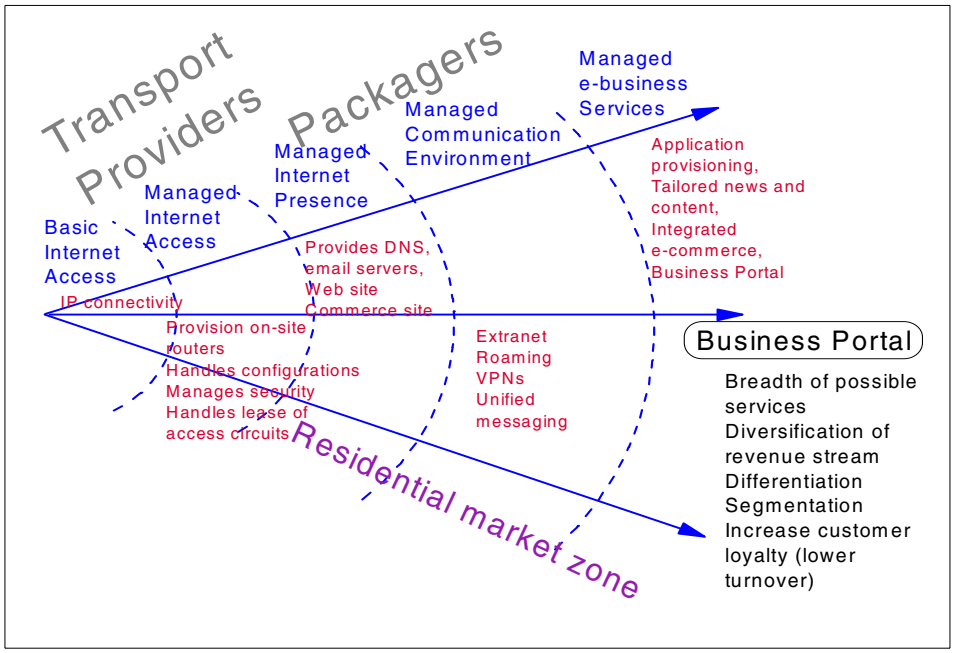


Figure 2. ISP business offer grid and residential mapping

1.2.5.1 Basic Internet access

The ISP provides only POP and IP connectivity to customers. Customers have to take care of a leased line to the POP and all IP and name allocations. They are also accountable for providing all web-related applications. This type of offer can only be targeted to medium and large businesses. It requires indeed significant technical expertise to manage infrastructure and administration.

1.2.5.2 Managed Internet access

The ISP provides another level of service by taking care of leased line connection, provisioning on-site routers, and handling configurations. This type of offer is only suitable for small businesses if they have the right technical skills.

1.2.5.3 Managed Internet presence

The ISP provides basic Internet services, such as Domain Name Server (DNS), e-mail servers, and web servers. This is the basic access for residential customers.

1.2.5.4 Managed communication environment

The ISP provides additional security features, building an extranet with the customer. It also provides roaming access by extending POPs around the world. This area is of less impact for residential users, specifically dial-up users. New technologies that have been deployed, for example, Digital Subscriber Line (DSL), provide a Virtual Private Network (VPN) connection between users and the ISP. The distinction between these types for business and residential is blurring. With time, the need for a more secure environment will be a requirement for residential users.

1.2.5.5 Managed e-business services

The ISP provides additional services around personalized content, e-commerce and collaboration utilities. Specific applications delivery is the logical extension once the ISP infrastructure is deployed. Applications can be made available for businesses, especially Small and Medium Businesses (SMBs), who can lower down the cost of maintenance of their infrastructure and logistics. This type of service will inevitably find an opening on the consumer market.

1.2.6 Services

Services are not the same for residential and business consumers. The requirements for businesses are much more difficult to meet in terms of performance, availability and most importantly security. The differences are illustrated in Figure 3.

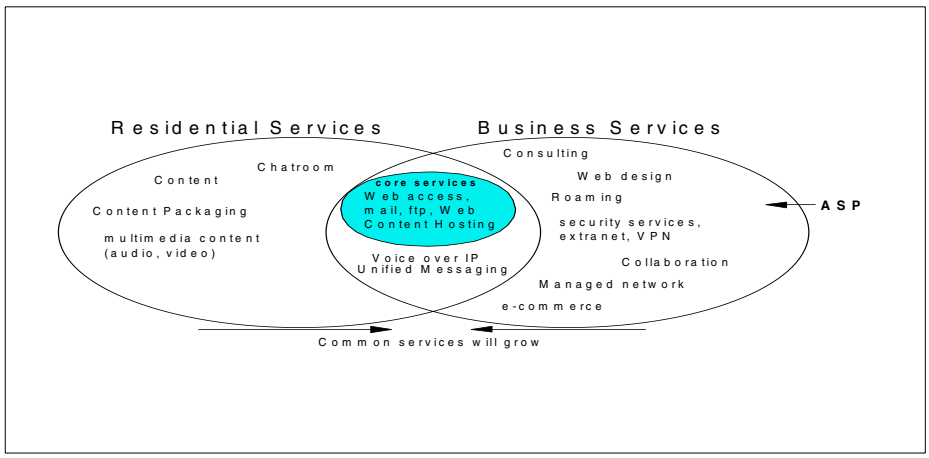


Figure 3. Core and value-added services

There is a shift on the mapping of these services. Common services and core services tend to grow over time. Business users requirements are leading the escalation in technology. After some time, that technology can be found on the consumer market. This push will inevitably benefit residential consumers as new innovating services and quality of service become available.

1.2.7 Internet Data Centers (IDC)

IDCs do the best work in the *Hosting Services* segment, as defined in Section 1.2.4, “Segmentation” on page 4. We think that RS/6000 SPs are well suited to fit in this type of infrastructure. The list of the products that run on a RS/6000 SP only emphasizes the complexity of delivering effective IDCs. This redbook demonstrates the value of a centralized management delivered by the RS/6000 SP. Centralized management can be key to run an IDC.

IDCs can be used to deliver a wide range of services that support e-business infrastructure needs. Some may be offered on a retail basis directly to e-businesses. Some may be offered on a wholesale basis to other service providers who will then resell the service either alone or, more usually, bundled with other value-added services. Some may be offered on both a retail and wholesale basis.

Typical services delivered from an IDC include:

- Co-location services
- Network services
- System management support services
- Managed storage services
- Content delivery services
- Managed web hosting services
- Managed application hosting services

IDC delivered services for an e-business infrastructure is shown in Figure 4 on page 12.

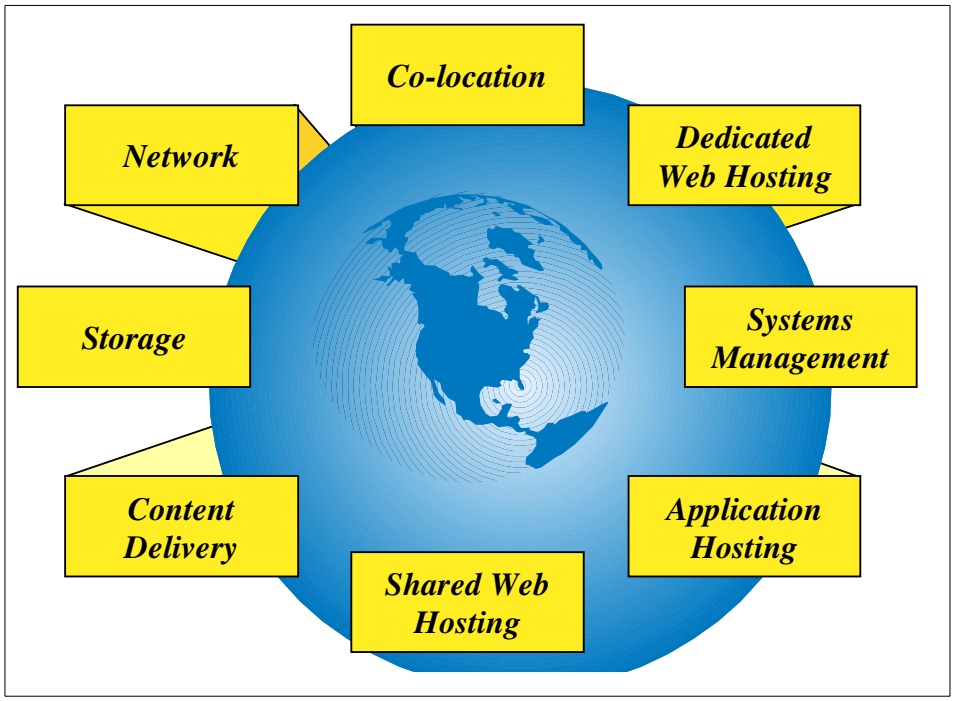


Figure 4. IDC delivered services for e-business infrastructure

Co-location services

The core service delivered from most IDCs is co-location. Some service providers may choose not to offer unbundled co-location services and will only offer co-location bundled with other services.

Basic co-location services usually include floor space and rack units where the co-location customer can install any e-business infrastructure of their own. They are located in a secure data center with reliable power and air conditioning. Customers pay a subscription fee based on how much floor space or how many rack units they require to house their equipment. Customers may have the option to have their equipment located in a secure cage or vault that is only accessible to the customer's personnel.

Co-location services are often sold on a wholesale basis to other service providers who will then install their own e-business infrastructure to offer value-added services. These services may compete with or complement the other services offered by the service provider who operates the IDC.

Network services
Most service providers will bundle some form of network service with their basic co-location services. Internet access is usually the major network service offered. Customers are usually provided with one or more LAN interfaces that can communicate with any other Internet attached device (usually 10Mbps, 100Mbps or 1Gbps Ethernet interfaces).

A variety of tariff structures may be used, such as:

- Peak bandwidth with, for example, x$ per Mbps
- Actual usage, such as y$ per GB transmitted and/or received

Some service providers may offer the capability for customers to connect with an ISP of their choice from the IDC. This is attractive for customers who want Internet access diversity.

Some service providers offer the capability for customers to have private network links to their own premises to facilitate remote systems management or connectivity to e-business infrastructure located on their own premises. Such circuits may also be required in the delivery of ASP services from the IDC or by wholesale service provider customers.

System management support services
Most service providers offer a range of system management support services to help their customers remotely manage e-business infrastructure located within the IDC. Typical management services offered include:

- Rebooting servers on a scheduled basis or in response to a failure
- Monitoring network capacity utilization
- Monitoring availability of applications
- Monitoring performance of applications
- Load balancing of incoming traffic across multiple servers supporting the same application
- Backup and/or restore of system and application data and offsite archival of the data in a separate secure facility
- Firewalls, intrusion detection and other security services
- Escalation of problems to the customer and/or the customer's designated equipment maintenance provider

Service providers that operate multiple IDCs may also offer the capability to perform load balancing across servers located in different IDCs or to route incoming requests to servers in the "nearest" IDC.

Managed storage services

Many service providers are now offering managed storage services to complement their co-location, network and system management services. Managed storage services provide a robust and reliable storage infrastructure that customers can use as an alternative to installing and managing their own storage infrastructure. Two types of storage service are usually offered:

- Storage Area Network (SAN) services provide one or more logical "disks" that are attached to the customer's servers via a SAN and replace directly attached disk devices.
- Network Attached Storage (NAS) services provide storage "capacity" that can be accessed from the customer's servers via a Local Area Network (LAN) or Storage Area Network (SAN) using industry standard protocols.

Managed storage services usually include backup and redundancy, for example, Redundant Array of Independent Disk (RAID), and can also support data replication, if required for e-business applications that must be always available.

Content delivery services

Service providers who operate multiple geographically distributed IDCs or who have a presence in multiple IDCs operated by other service providers can offer content delivery services. Content delivery services provide two major capabilities:

- Distribution of content to storage facilities located at multiple IDCs
- Routing of requests for content to the storage facility located in the IDC that is capable of delivering the content with the desired level of performance

Such capabilities simplify the effort required to deploy content-rich e-business applications in a manner that can provide high levels of performance to end users located in multiple geographic areas.

Managed web hosting services

Many service providers support managed web hosting services that are targeted at customers who want a more comprehensive web hosting infrastructure than the combination of the more basic IDC services can provide. Managed web hosting services are usually categorized as:

- Dedicated

  The customer has their own dedicated web server(s). These are often procured on behalf of the customer by the IDC.

- Shared

  The customer shares a web server with other customers. When the application is shared, the most important issue for customers is data isolation (from other customers). This must be seriously addressed by the IDC.

Managed web hosting services usually include the installation and management of both the operating system and web server application environment on each server by the IDC and will often leverage other services provided by the IDC, such as storage management and content delivery.

Managed web hosting services are often complemented by professional services to assist the customer in the design and implementation of their web presence.

Managed application hosting services
Managed application hosting services target the emerging ASP market. It provides hosting of common business applications for use within a business, by its customers, its suppliers, and other business partners.

Service providers often deliver these services from their IDCs in partnership with application providers who own the applications that are being hosted by the service, for example, an Independent Software Vendor (ISV).

In concept, managed application hosting is similar to managed web hosting, but there are several important differences:

- The range of applications that must be supported is usually much larger.
- The Service Level Agreements (SLAs) are often more stringent.
- The network connectivity requirements are more complex, as they often involve non-Internet links or secure tunnels over the Internet.

Implementation of the application for a customer may require extensive professional services to integrate the application with the customer's business processes.

1.3 Requirements for ISPs

We have divided the ISPs requirements into three different parts:

- Functional requirements describe functionality requirements for ISP businesses and ISP platforms.

- Non-functional requirements come from the business side and these requirements provides basic figures to define architecture for ISP platforms.
- Architecture evaluation criteria are generic principles to make the architecture work and define the platform than can answer businesses' requirements.

1.3.1 Functional requirements for ISPs

The ISP platform described in Figure 5, is comprised of three parts:

- The ISP functions part which contains the main functions of the ISP services. For example, forum, content management, and instant messaging can be used for the community group service.
- The infrastructure services part, which contains all the components providing services for the platform.
- The administration services part, which contains the administration functions of the platform.

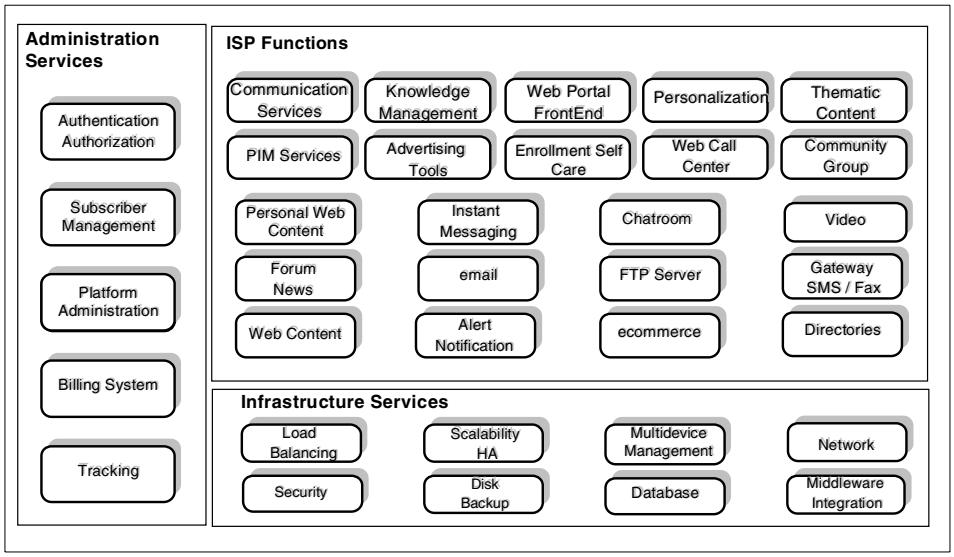


Figure 5. Logical architecture - components of an ISP

A solution is based on the following logical components:

ISP functions

- The *advertising tool* is composed of a web advertising tool that is a server sending a banner of advertising to the Web server according to

parameters (size of the banner and user group) put into HTTP requests. This tool is also able to administrate banners and collect and compute the number of accesses per banner. Additionally, a broadcaster advertising tool can send targeted advertising to groups of users or companies through several types of communication means, like e-mail, SMS, or Voice.

- The *web call center* provides the tools for customer self-care that supports customers via a Web interface. The customer can ask questions to operators and find information about the portal functionality.
- The *web content portal* is the frontal interface for all users to access all the portal services. The services can be personalized according to the user profile and the type of device. The portal must contain a search engine to help users navigate within the portal domain.
- The *knowledge management* component is responsible for providing added value on the information collected by tracking components. This source of information can be used, for example, for marketing purposes, data mining, FAQ, or statistics; the goal is to improve the content of the site to match the preferred content of customers.
- The Personal Information Manager (*PIM)* personal space component provides personal functions like an agenda, an address book, favorite URLs, and direct links to specific content.
- The *community group* is a space to exchange ideas and information for a specific group of users sharing common interests.
- The *communication services* is an interface to access all the standard communication services like e-mail, chat, or instant messaging.
- The *enrollment/self care* component provides functions to enable users to subscribe to the platform and/or modify their account information.
- The *thematic portal* is comprised of different parts per pool of interest.
- The *personalization* function provides personalized services according to the user profile. The user needs to be authenticated and his access rights identified. These two steps are a mandatory part of the personalization process.
- The *directory* contains the address book of the users. This is the white pages of the customers, which can be used to publish customer information for other services on the platform.
- The *mail system* can be accessed both through the web or from a mail client. The mail system is interfaced with the subscriber management and automatically creates mail boxes when a new user subscribes to the

portal. This is the service activation process, followed by the provisioning of the mail account.

- The *personal web content* component is a web application which receives static web pages edited by the users and publishes them on the web. This application is linked to the subscriber management to open a space for new users and check the rights of each user.
- The *forum/news* is a server which gives access to the standard Usenet news (NNTP) of the Internet.
- The *web content* component contains general information about the portal.
- The *chatroom* is a server that gives access to the Internet Relay Chat (IRC) standard services of the Internet.
- The *instant messaging* is a new community group service.
- The *FTP* server allows the user to download material such as software, catalogs, audio files, and video files from the site via FTP.
- The *fax/SMS gateway* provides the ability to send fax and SMS messages.
- The *video* component provides the ability to display video for education, training, entertainment, or collaboration, among other possible reasons.
- The *alert/notification* component allows notifications to be sent to users when an event occurs on the platform, for example, a new entry in the calendar.

Infrastructure services

- The *load balancing/scalability* function allows an easy service scalability that copes with the growth of traffic, and also actively contributes in building a High Availability (HA) configuration.
- The *security* function is responsible for protecting the platform and the sensitive data against Internet hackers.
- The *multi device management* function is responsible providing content according to the type of the device, such as Wireless Application Protocol (WAP), Personal Computer (PC), screen phones, and Personal Digital Assistant (PDA).
- The *disk/backup* system contains all the data of the platform shared by all services. The backup system helps ensure the reliability of the platform in case of a disaster.
- The *database* contains all the data of the platform shared by all services.

- The *network* is the infrastructure. It is the backbone of all connections between the machines of the platform and the connection with the existing networks that form the Internet.
- The *integration middleware* component enables the integration of all databases within the system.

Administration services

- The *billing system* is the component that runs the billing system engine. It is able to compute and send fees to users according to their type of subscription.
- The *subscriber management* component contains the subscribers' database. This database is the central repository of the platform and provides users' data to other elements of the platform. It also contains all the user profile information. This function provides more added value, such as personalization and classes of service. A method of access to this server is with the Lightweight Directory Access Protocol (LDAP); conveniently, this is a LDAP server. Finally, the subscriber management delivers (among others) the following applications:
  - Subscriber on-line enrolment
  - Content personalization server, which is used to serve personalized welcome pages
  - Customer care application for hot-line services
  - Authentication manager
  - Administration and statistics reporting tools
- The *authentication/authorization* services component is responsible for identifying users and delivering users' access rights. Authentication enables personalization of services and is used by all services to identify users. Authorization checks to see if they are authorized to access services. Authorization should be centrally managed, thus providing single sign-on to all services.
- The *platform administration* component allows the administrator to control the behavior of the platform and take administration actions (if needed). The interventions can apply both to software components and hardware components of the system. It involves activities such as:
  - Adding and removing users and systems.
  - Monitoring and executing computer and network performance tuning.
  - Updating of application programs such as adding new functions and changing data formats.

- Fixing bugs.
- Adapting the software to new hardware devices.
- Distributing and updating application and data files.
- Hardware and software diagnostics.

- The *tracking* component monitors all the events that occur on the platform and also collects the information about the users behavior for the Knowledge Management system and for the marketing organization of the portal.

1.3.2 Architectural evaluation criteria

The ISP architecture evaluation criteria are generally defined principles and requirements for ISPs' architecture definitions and implementations. We take, as a basis for our discussion, a front end (DeMilitarized Zone or DMZ) and back end (trusted network or secure zone) architecture. This is a deployed, distributed architecture that improves the crucial features of the ISP platform. A representation of back end/front end architecture can be found in Section 1.4, "An architecture umbrella with seven zones" on page 24.

Number of subscribers
The number of subscribers targeted defines the size of the implementation of the ISP infrastructure. This always comes as a difficult requirement to set, but it is essential to start any configuration. Many factors come into play to define that number, such as budgetary limitations, the number of subscribers in current business activities, and the number of subscribers from pre-deals with businesses.

Number of simultaneous users
The number of simultaneous users which are supported is defined on the total number of subscribers handled by the ISP. These numbers are estimated from known behavior analysis. For example, business users will most likely read their mail in the morning, while residential users will read theirs in the evening. The peak hours are not the same.

Estimates help build the maximum number of users that require a specific service. The global estimated peak hour connections define the bandwidth required to provide performance services and define the type of machine required.

Ease of use
Avoiding turnover is one of the number one goals for ISPs. One of the reasons users change from one ISP to another come from the difficulty of using content and services. Making an easy-to-use environment for users, is

to create a stable interface with them. Presentations that change too often can lose non-expert users. Personalization of content and services is also a way of improving usability of the service. Finally, the ability for users to personalize their own portal pages is a must.

One of the latest challenges is to deliver services and content to a multitude of devices that range from traditional personal computers to Personal Digital Assistants (PDAs) or Wireless Application Protocol (WAP) phones. New devices have smaller screens and limited input capabilities. The ISP platform has to deliver the same Quality of Service (QoS) and content to all devices.

The architecture of the platform must enable all these criteria to be implemented in a controlled and easy manner. Complexity in numbers must not impact simplicity of deployment and maintenance on the platform.

Ability to add new device / network type
As said previously, new devices have generated a more complex landscape for ISPs to handle. New networks with new protocols are used to connect to the Internet. Even as we speak, new technologies are been devised.

The ability of the ISP platform to accept those changes as they come along is a necessity. More importantly, the platform must evolve, not change, with these additions, as changing the infrastructure to adapt is not an option. Therefore, new device types should be easy to add and it should be possible to describe, manage, and access device types centrally. New protocols should be easily adapted in a “plug in” addition fashion to software components on the platform. Interruption of service should be avoided.

New mobile devices also add new behaviors, such as the necessity to replicate information to devices that can function in a standalone fashion.

Ability to add new services and content
The ISP platform needs to be flexible enough to accept adding and removing services and content without interrupting services. Flexibility is greatly enhanced by multilayer, front end, and back end architecture. On the front end of the platform, the ideal situation is to have a single function per server or node in order to have the maximum granularity of services. Adding or removing a service is adding, removing or disabling a server or a node.

Adaptability
Upgrading platform components or changing them should not generate service interruption. At the very least, this should be achieved in a controlled, time boxed way. The platform should permit this to happen.

Performance

The performance of the platform is constrained by the type of access employed by the user. Whether the user is connecting through a modem or through a leased line will have some obvious consequences on the performance of the services delivered. In all cases, the platform must not be the weak link in this chain of delivery to the user. The hardware and software components should be designed in order to deliver the best response time on the best network connection available. Another advantage of designing the ISP platform in layers and isolating services on different servers is the possibility of controlling the performance of the platform with a very small granularity and in a very controlled manner.

The user will understand a slow service delivery through modem connection, but will unlikely accept bad performance through other larger band accesses. Services must be tuned in regardless of the type service delivered. For example, simple services should be delivered in the range of a few seconds, even through modem connections.

Scalability

Scalability of the platform is the platform's ability to adapt to a growing number of users. Of course, scaling must be done with as few interruptions of service as possible. ISPs often have to face that type of operation, as it is usually difficult to predict how successful its services will be on the market. Therefore, ISPs usually start off with a small configuration and then evolve as new subscribers sign up.

Scalability applies to all layers in the architecture of the platform and in many other ways. Scalability is best achieved with front end/back end architecture.

Front end servers can be scaled horizontally and vertically. Horizontal scaling is the process of adding a new server or node that delivers the service that needs scaling. Vertical scaling is achieved by increasing the existing server or node memory or CPU power. Not all machines have that capability, and they should be chosen carefully in order to enable that kind of improvement. Horizontal or vertical scaling should also not interrupt services. This is made possible by using the load balancing technique, which guarantees the service is still available on a server while a scaling operation is performed.

Back end servers can be scaled vertically. Back end servers usually host data access servers that run in a single instance. Clustering techniques need to be used to assure the scalability of these servers without long interruptions of service.

High availability

The availability expectations of the system are related to how many hours in the day, days per week, and weeks per year the content and services are going to be available to the users. High availability is not only achieved by a good architecture layout, but also by the software components features and functions delivered.

The first requirement in achieving high availability for an ISP is ensuring there are no single points of failure in the architecture. This means that there should be duplication of all services available and automatic recovery from failures on single instance applications in the system. This also means duplicated network lines. These duplications are in themselves a challenge to put in place and to test.

The second requirement in achieving high availability for an ISP is to devise plans for recovery in case of unexpected failures, human errors leading to failure, or disaster. This is achieved through a rigorous planning of human resources, backups and operations. This is done with adapted ISP organization.

The third requirement in achieving high availability for an ISP is to deliver the right level of service to customers. This is pledged through a Service Level Agreement (SLA) with users, which defines the quality level of service the ISP should deliver at all time. This is done by providing help desks and technical support desks through different channels, such as telephone, mail, web forms, or e-mail. Once again, the ISP organization is greatly impacted by such operations, which can be heavy and difficult to implement, especially for smaller ISPs.

Security

The ISP services and content must be deployed in a platform that provides authentication, access control, and auditing capabilities to ensure the integrity and privacy of business-critical information.

Splitting the various services and functions on different servers provides key advantages from a security perspective, because the appropriate level of security can be assigned to each server based on its role. In general, ISPs will implement this security by designing their IP network with different subnetworks and interconnect them through firewalls and routers configured to provide just the minimum required level of access to a given subnetwork.

1.4 An architecture umbrella with seven zones

Regardless of which type of market a service provider targets, they must position themselves for growth and capability in order to implement swift changes in their business infrastructure and IT architectural design. An e-infrastructure that meets the demand of new business models and that can handle an increasing number of subscribers, additional services, challenging workloads is critical for success.

The criteria for architectural design, as described in Section 1.3.2, “Architectural evaluation criteria” on page 20, have motivated IBM systems architects to propose a service provider architecture around seven functional areas, which are shown in Figure 6 on page 25. It is not a generic all-in-one platform, which has no reality versus the diversity of requirements one service provider has, but it offers agility for a constantly evolving environment, making it easy to change components, products, or modules.

The seven zones of an e-infrastructure, represented in Figure 6 on page 25, can be found at `http://www.ibm.com/e-business/infrastructure/uk/zones`.

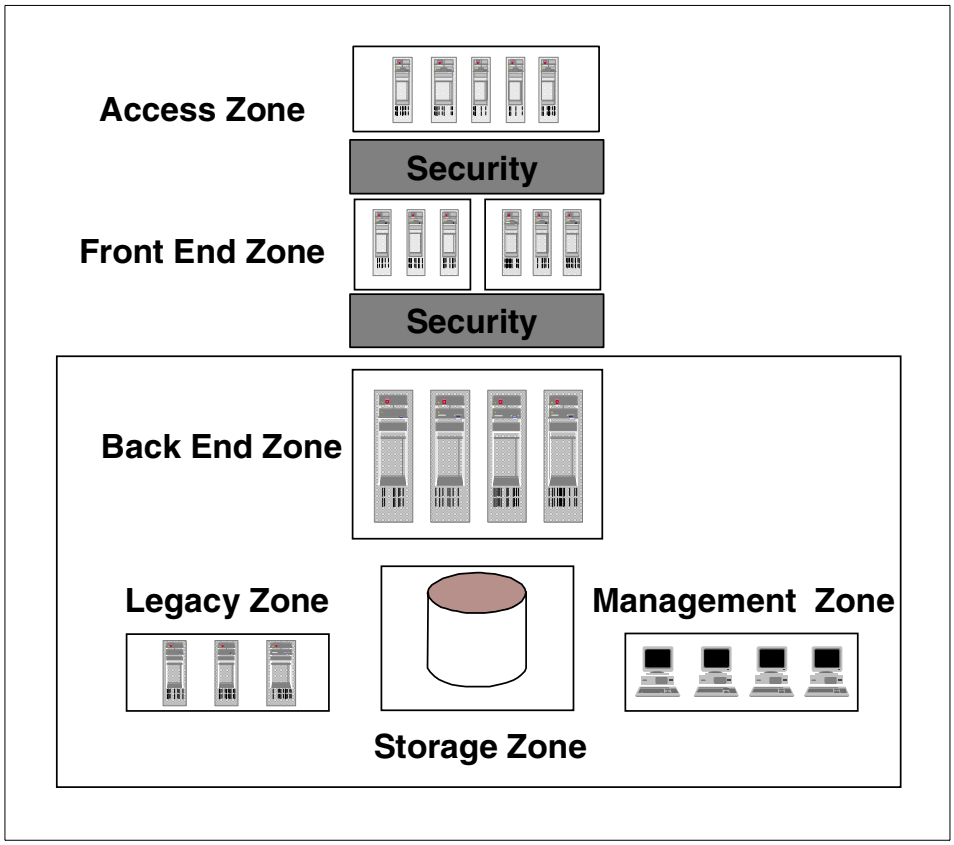


Figure 6. Seven zones architecture

1.5 Scope of this publication

The focus of this redbook is on *Packagers*, as defined earlier. These are ISPs that provide services from Managed Internet Presences to Managed e-business Services, as described in Figure 2 on page 9. Even if this is of interest for all geographies, the emphasis is made on how value added services can be used to make a competitive difference on the ISP market.

In the next chapter, we will define what the required components of an ISP today are.

Chapter 2. Overall architecture for Internet Service Providers

This chapter gives an overview of an ISP environment. It provides the architectural principles aimed at understanding this environment and shows some guidelines for building an architecture definition. Functional architecture is mapped onto functional requirements, which were defined in Section 1.3, “Requirements for ISPs” on page 15.

The ISP landscape outlines the basic structure for this chapter. In this chapter, we explain:

- What is the Internet network outlook and main components.
- Where an ISP platform is located on the network.
- What devices can access an ISP and how they access that ISP.
- What basic services an ISP can deliver.
- What will an ISP have to do to get started.
- What are the principles for implementing security.

The overall architecture for an ISP based on its functional architecture is shown on Figure 6 on page 25. The ISP functional architecture is divided in seven zones which are:

1. Access zone

   The access zone provides the connection by which services may be used by both the Internet and analog, ISDN, cable, wireless (such as GSM and WAP dial up), and fixed line users, and is connected to the security zone. The access zone is described in Section 2.1.2, “Access network” on page 38. It provides channels to send and receive data. The access network provides the entry flow of the ISP platform. This entry flow enables communication by using different communication protocols between nodes that are implemented in a distributed way.

2. Security zone

   The security zone filters Internet traffic so that it proceeds only to access the front end zone. If the environment requires a high level of protection against intrusion, security zones are installed as concentric layers in the architecture. A more detailed description of security implementations can be found in Section 2.2.2, “Security layers” on page 58.

3. Front end zone

   The front end zone receives and manages users requests for services, such as mail, news, or Web. No specific techniques are required in this

zone to share or duplicate data. The front end zone is indeed a set of servers that run independently with virtually no coupling at all. These servers are usually called protocol nodes, as they are communication interfaces. Protocol nodes dispatch service requests to data access nodes. They do not contain user information.

The front end zone is on the DeMilitarized Zone (DMZ), which separates the Internet from the Secure Zone. A detailed description of the front end zone implementation can be found in Section 2.4, “Front end zone” on page 69.

4. Back end zone

   The back end zone is connected to the front end zone, management zone, and storage zone, and contains all servers with applications who work on user information. Servers running on the back end zone are called data access nodes. These nodes must be implemented on a trusted network or secure zone. This is what the separation brings in for securing the environment. Moreover, by clearly separating front end and back end functions, it is possible to apply the most appropriate techniques to achieve scalability and high availability on the ISP platform. A detailed description of the back end zone implementation can be found in Section 2.5, “Back end zone” on page 76.

5. Storage zone

   The storage zone manages the administration of all the disks and stores all user data and content in a multi vendor and platform environment. In an heterogeneous environment, where more than one type of server or Operating System (OS) needs to access data stored, a Storage Area Network (SAN) should be used. SAN also improves the performance of access to data, which can be critical, especially with large databases. SAN not only enables aggregate distributed storage, it also integrates backup solutions. A detailed description of the storage zone implementation can be found in Section 2.6, “Storage zone” on page 82.

6. Management zone

   The management zone is used to manage the complete network infrastructure, providing billing, logging administration, and customer services center information. The Management zone is addressed in Section 2.7, “Management zone” on page 87.

7. Legacy zone

   The legacy zone contains the business computers that already exist in an enterprise and contain valuable applications and data that are made

accessible through the Internet. The Legacy zone is addressed in Section 2.8, “Legacy zone” on page 94.

We describe, in the following sections, each of these zones.

2.1 Access zone

The network environment of an ISP is illustrated in Figure 7 on page 31. This figure shows the complexity of this environment.

The existing infrastructures are numerous and complex. Whether it is the Telco networks that are used to remotely connect users’ electronic devices or the Internet, which is a complex construct of networks, the ISP has to interface with them. This interface is what we call the access network.

The access network of the ISP has to cope with interfacing with a mouthful of protocols either generated by the device/network connecting or by network security protocols implemented. Each of these protocols are a service delivered by the ISP platform. Each of these services are described in this chapter.

First, we provide an overview of the layout of the existing infrastructures and how the ISP platform fits in. The platform has interfaces with all types of terminals. Terminals access the ISP platform via specific gateways, which are described.

We then analyze the process of connection creation and authentication. This process is the first end to end communication that occurs when setting up a session between the ISP and the subscriber. We take a look at all the interfaces that handle such process. These interfaces are the components of the access zone.

We next describe a detailed example of connection creation when using roaming access. In order for successful roaming among an arbitrary set of ASPs to suceed, the ASPs must use convergent methods to share:

- Dial-in numbers
- Connection management
- Authentication
- NAS configuration and authorization
- Address assignment
- Routing, security, and accounting

Detailing such implementation shows all the capabilities of the interfaces used.

We next take a look at Domain Name Servers (DNS). This is the second end to end communication that usually takes place after the connection creation. DNS are an essential part of the ISP platform. Indeed, the ISP platform has to perform intensive DNS resolutions. For performance reasons, the platform should run its own DNS server.

Finally, we take an intensive look at the network security. Security can be implemented at the lower stacks of the TCP/IP protocol to provide a secure framework of transmission. We show how a Virtual Private Network (VPN) can be implemented and who is responsible for implementing it. We refer back to Section 1.2.4, “Segmentation” on page 4, where we have defined the segmentation roles for ISPs:

- Transport provider: Provides POP to customers.
- Packagers: Provides services to customers.
- Hosting services: Provides hosting solutions to customers.

Transport providers and Packagers may require provisions for building VPNs with customers, depending on the type of offer they choose to provide:

1. Network infrastructure: Telco, Network Provider (NP)
2. Basic Internet access: Point of Presences (POP) and handle network configuration
3. Managed Internet access with basic services: ISP provides Web services, like DNS, mail, and Web
4. Managed communication environment: VPN, Roaming
5. Managed business services: ASP, commerce

This will be explored in further detail by looking at the scenarios in which a VPN may be required.

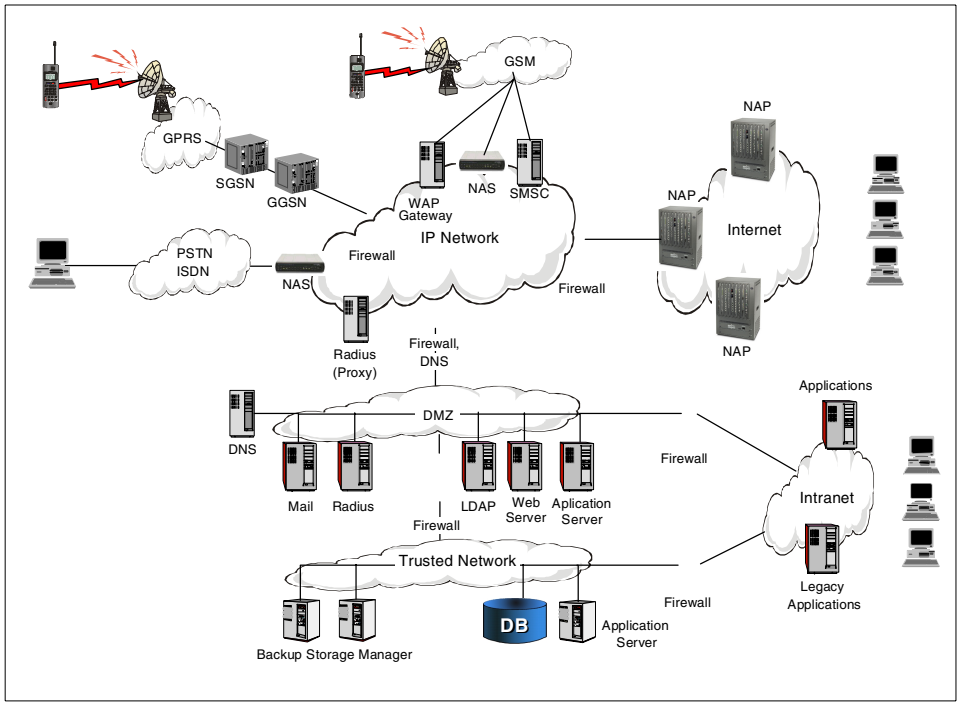


Figure 7. Overall network architecture

2.1.1 Network infrastructure

The network infrastructure is comprised of:

- Devices, through which end users access the network
- Telco networks, like PSTN, GSM and, in the near future, GPRS and further on UMTS
- POPs, provided by Service Providers to connect devices to the Internet and be part of it
- The Internet, built by LANs and WANs of businesses, organizations and service providers

On Figure 8 on page 32, the network infrastructure architecture is illustrated in a simplified way. The picture depicts how parties interact together:

- Users who can be connected with many distinct devices and through many distinct networks and protocols

- The ISP platform, which is really the interface between the Telco networks and the Internet
- NSPs or Transport Providers, who provide connections with NAPs to all networks who want access the Internet
- Third parties, like other ISPs, portals, Application Service Providers (ASP), who provide added value services and legacy systems to ISPs, and through the ISP, services to the user

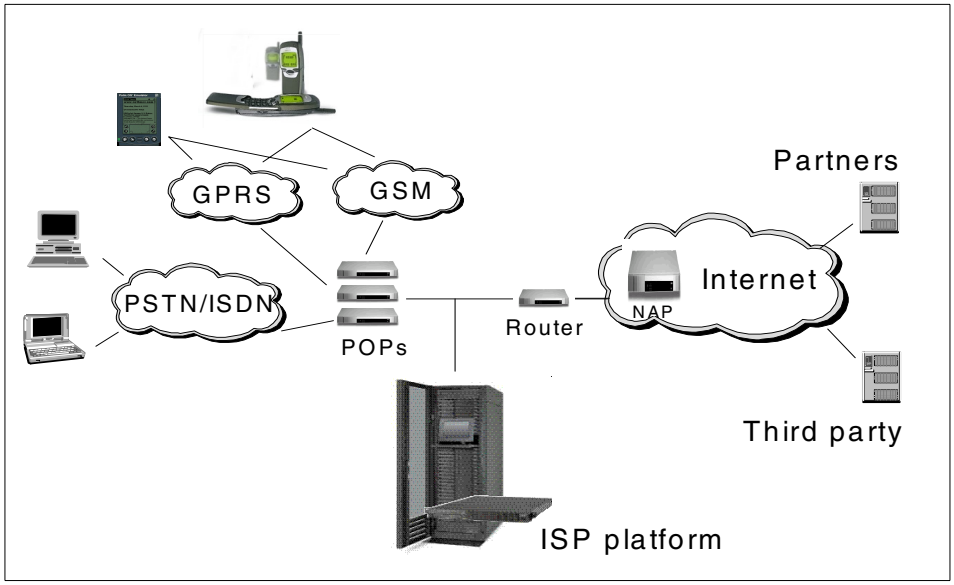


Figure 8. Network environment for an ISP

The following describes each component and provides a more detailed view of their sub-components. We also provide an overview of the evolution, in terms of technology and performance, of the network infrastructure.

Devices

Surprisingly enough, the best way to categorize devices that can access the Internet is not by splitting them by physical type or network infrastructure access type. It is indeed normal to talk about PCs, mobiles, PDAs or dial-up access, cable access, DSL access. The best way to categorize devices is through the protocol they use to connect. The protocol defines the requirements the device has to meet and the operations the device will be able to accomplish.

Devices can be divided in two categories:

- Thin client devices

  Thin clients are devices with no software installed. For example: terminals (in the traditional sense), net computers which were the trend a few years ago, WAP devices, or screen phones. The Web browser falls also in this category. Only configuration parameters are stored on these devices.

- Fat client devices

  Fat clients are devices with application software installed. For example: PCs or PDAs.

There is an interesting discussion around the choices for one or the other model of device. Fat clients do provide the ability to work offline, because applications are always available. Nevertheless, fat clients will tend to be more expensive, as they require storage, greater CPU power, and more memory. Thin clients also require less to no maintenance.

More interestingly, this model applies to Service Providers who can deliver applications to consumers (the ASP model), gradually transforming the fat client PCs into thin clients through the use of the Internet. The question is then: are fat clients going to disappear? Are applications going to be fully centralized in Service Providers and accessed through the Internet? We will surely see more of this in the next few years, but fat devices still have a long life.

Existing devices will also have access to the net with technologies like *Bluetooth*. The Bluetooth wireless communications technology is a radio-based standard for connecting and exchanging data between devices. Promising to be cheap, small radio devices may be inserted in all existing appliances ranging from mobile phones to television sets to refrigerators. This can be seen as a Wireless Personal Area Network (WPAN) of home and personal electronic devices that could then be extended to the Internet.

PSTN

Public Switched Telephone Network (PSTN) is the international telephone system based on copper wires carrying analog data. PCs and laptops use a modem to connect to the PSTN. A Network Access Server (NAS), which is a Point of Presence (POP) for the ISP, converts the communication into the IP protocol, enabling connection to the Internet and the ISP services.

The newer telephone networks are based on digital technologies, such as Integrated Services Digital Network (ISDN) and Digital Subscriber Line (DSL).

For ISDN, PCs and laptops use a digital modem, which has a much faster data exchange rate.

For DSL, PCs and laptops use a network card (Ethernet or ATM) connected to a modem/router component. This modem/router is then connected to a modulator that is plugged on the telephone line. Thus, the data flow is modulated over the telephone line. Unlike traditional modems, the telephone line is still free to send a receive calls and therefore there are no charges associated with connection time. Moreover, DSL customers are usually within a VPN, where Point to Point Tunneling Protocol (PPTP) can be used.

Let us use Asymmetric Digital Subscriber Line (ADSL), which is one of the most popular DSL, as an example of DSL. ADSL modems use digital coding techniques, which make it possible to squeeze up to 99 percent more capacity out of a phone line. ADSL modem technology really transforms ordinary phone lines into high-speed digital lines for ultra-fast Internet access. ADSL provides speeds up to 8 Mbps downstream (from the Internet to the user) and up to 1 Mbps upstream, depending upon line length and loop and line conditions.

DSL is suitable for residential and business users, which means being in a VPN has some importance for businesses. High performance and endless connections are always a plus for any consumer. High bandwidth is mandatory in that context to deliver seamless services.

From an ISP standpoint, this type of technology addresses requirements from both residential and business consumers at the same time. High performance opens new market opportunities for broadband billable services, such as audio and video streaming. From a consumer standpoint, it is the next best thing to a leased line connection.

Figure 9 on page 35 illustrates this PSTN environment.

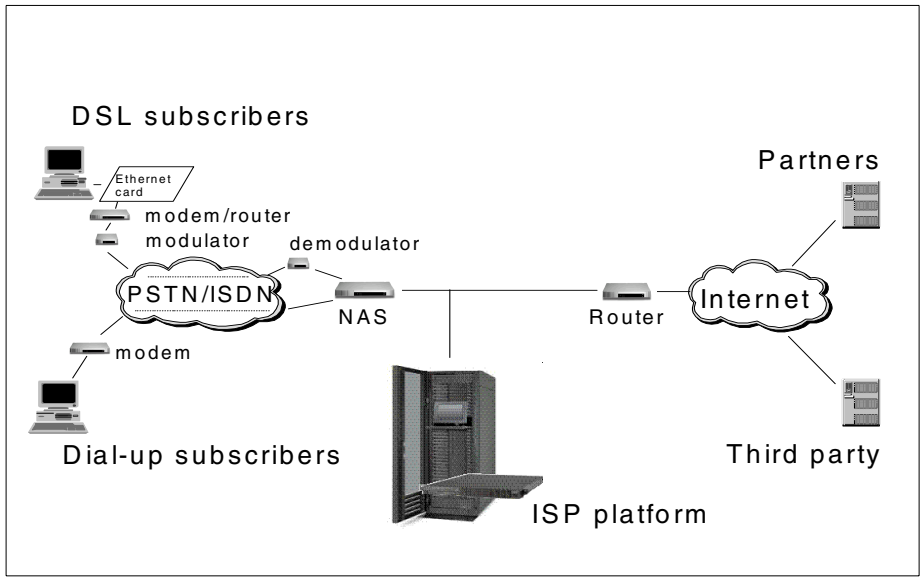


Figure 9. Access through PSTN

GSM

Global System for Mobile (GSM) communications is a circuit switched mobile network. The GSM network creates a platform for wireless wide-area data communication by offering a variety of data services (both bearer services and teleservices). GSM specifies two asynchronous bearer services: transparent and non-transparent. The non-transparent mode is implemented using the Radio Link Protocol (RLP). It is more reliable than the transparent mode. The bit error rate is required to be less than 10^{-8}. The maximum line rate is 9600 bps.

The price paid for the low bit error rate is the occurrence of highly variable transmission delays. Under unfavorable conditions, the delays can extend to tens of seconds or minutes. Conditions may become so unfavorable that a mobile telephone transmission path could be temporarily broken.

GSM makes the wireless access to fixed data networks possible almost everywhere in Western Europe and also in some countries outside Europe. GSM specifies interworking with several wireline networks, including PSTN and ISDN.

Wireless gateways are required to transform protocols into TCP/IP mode. These gateways are specifically built to handle local communication and caching to the GSM network.

Short Message Services (SMS) is part of the GSM standard. It provides message services to mobile users.

Figure 10 depicts this GSM environment.

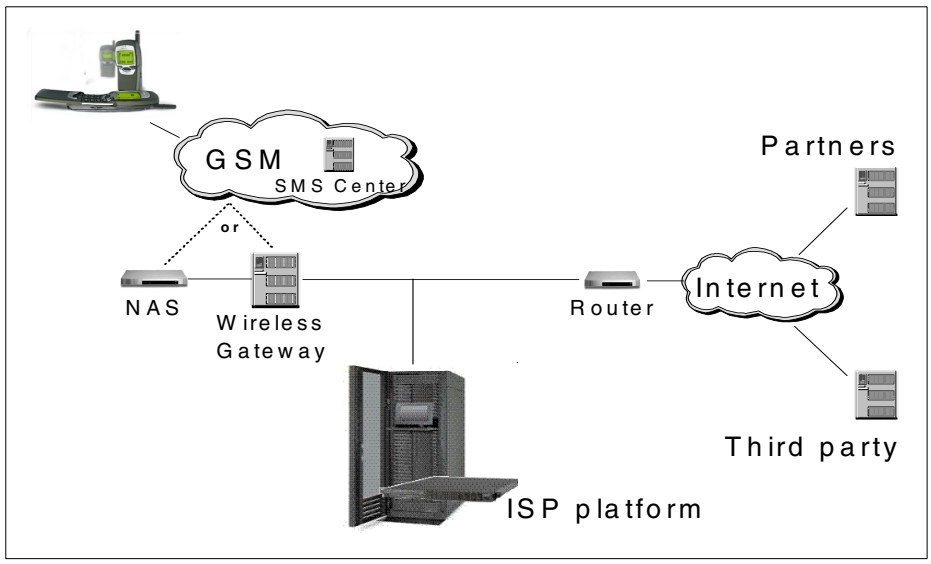


Figure 10. Access through GSM

GPRS

General Packet Radio Service (GPRS) is a new standard for wireless communications. This new nonvoice value added service allows information to be sent and received across a mobile telephone network. It supplements today's Circuit Switched Data (CSD) and Short Message Service (SMS). Theoretical maximum speeds of up to 171.2 kilobits per second (kbps) are achievable with GPRS. GPRS facilitates instant connections whereby information can be sent or received immediately as the need arises. Connections are subject to radio coverage, but no dial-up modem connection is necessary. It is based on packet switched technique.

GPRS facilitates several new applications that have not previously been available over GSM networks. This is due to limitations in speed of CSD (9.6 kbps) and message length of SMS (160 characters).

The GPRS network is implemented by the GSM network infrastructure, adding two network nodes for the GPRS backbone. The GPRS backbone enables communication between the GPRS support nodes and is based on a private IP network. The GPRS network includes two different support nodes; these are the Serving the GPRS Support Node (SGSN) and the Gateway GPRS Support Node (GGSN). The main functions of the SGSN are GPRS mobile authentication, registration and mobility management. GGSN works like a router and is the interface to the ISPs data network.

The mobile customer uses a WAP mobile with a micro browser, a mobile with SMS services, and/or a GPRS terminal. The GPRS terminal connects to the ISP platform via Telcos GPRS network.

Figure 11 depicts this GPRS environment.

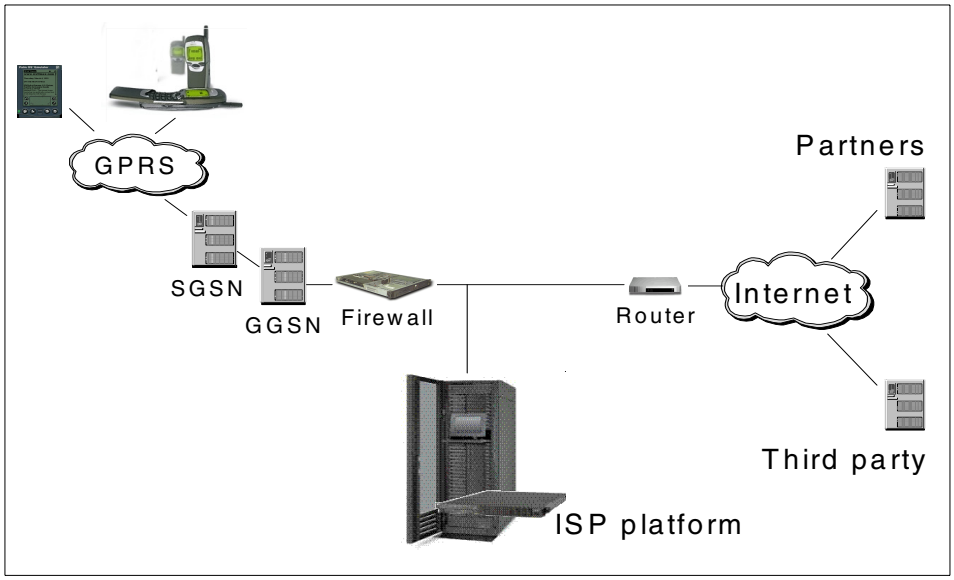


Figure 11. Access through GPRS

POP

POP is a telephone number that gives users dial-up access to the Internet. Devices that provide access are generally NASs or components providing a wireless gateways.

The telephone number is usually regional, but can be national. These national telephone numbers are usually charged as regional telephone numbers. ISPs provide many POPs to their customers, even international

POPs. These are the requirements for roaming access, as described in Section 2.1.3, “Roaming between ISPs” on page 43.

Router
Routers are the components of the network infrastructure that handle the routing of the packets between the network interfaces. Routers maintain a dynamic list of Internet addresses, maintaining a dynamic picture of the topology of the network. Furthermore, routers maintain counters for performance, resource use, link use and link conditions. They are usually maintained remotely for easier and centralized administration. Finally, routers modify the basic action of the forwarding process based on source address, destination address, and protocol type by adding packet filtering functions. Routers are often used as network access firewalls because the solution is cheaper, easy to maintain, and has a high performance.

Internet
The Internet is a global network interconnecting billions of computers and devices. The Internet is decentralized by design and each host is independent. Any computer or device becomes part of the Internet as soon as it gets an IP address. A mobile phone connected via an ISP to the Internet will become part of it.

The Internet infrastructure is no more than a loose organization of all networks interconnected together, with the networks being LANs and WANs of businesses, organizations and service providers.

NAP
Network Access Point (NAP) allows ISPs to interconnect and exchange information among themselves. NAP is an exchange point for Internet traffic. NAPs are interconnected through very high speed backbones. These interconnections represent the skeleton of the Internet. ISPs connect their networks to the NAP to exchange traffic with other ISPs and third parties on the Internet. The exchanging of Internet traffic is generally referred to as peering.

NAPs are traditionally handled by Network Service Providers or Transport Providers.

2.1.2 Access network

The ISP platform is interfaced with infrastructure networks like PSTN or GSM, and the Internet, through external interfaces. These external interfaces define what we call the access network of the ISP platform.

Registered users of an ISP need to be authenticated. For example, users may connect to the ISP platform through the Telco networks to gain access to the Internet and eventually to services delivered by the platform. Users may also connect to the ISP platform from the Internet to gain access to the platform services. One of the functions of these external interfaces is to enable authentication.

These external interfaces are standard interfaces. We have identified these external interfaces for network infrastructure and access network integration:

- NAS and authentication servers
- RADIUS proxy - RADIUS server
- Short Message Service Center (SMSC) - ISP platform
- GGSN - Authentication server
- Proxy - Authentication server

2.1.2.1 NAS and authentication servers

The NAS interfaces with an authentication server during the dial up connection creation and authentication process. The authentication server can be, for example, a RADIUS server or a TACACS server. A detailed description of the authentication process is provided in Section 2.1.3, “Roaming between ISPs” on page 43.

NAS

The NAS is a device that has interfaces both to the backbone and to the ISP IP network and receives calls from hosts that access the IP backbone through dial-up services. A NAS is located at an ISPs Point Of Presence (POP) to give Internet and ISP platform access. NAS units are traditionally a combined modem and protocol-based access server. Modem units are configured in a PSTN rotary group. Today, ISDN services are getting much stronger. Point to Point protocol (PPP) is used to establish the IP access session.

The Remote Authentication Service (RAS) is executed as a part of the Point to Point Protocol (PPP) and authenticates the phone connection between the mobile and the PPP peer. The PPP peer is the NAS; concretely it is the modem.

RADIUS

RADIUS is the most common type of authentication protocol used. RADIUS solves the problems associated with meeting security requirements of remote computing. RADIUS is a system of distributed

security that secures remote access to networks and network services against unauthorized access.

A RADIUS Client must be present in the IP network. This RADIUS client could be in charge, depending on the policy of the ISP, of allocating IP addresses to terminals when connected. The Radius client works in conjunction with the NAS, which will support, for example, ISDN connection setup initiated from the terminal.

RADIUS includes an authentication server and client protocols. All ISP user authentication and network service access information is located on the ISP's RADIUS server. The RADIUS server can be located on the IP backbone or/and on the DMZ.

RADIUS Proxy is an optional component of the transport network. It handles the RADIUS protocol with the client side NAS and relays the requests to the appropriate ISP identified in the login identifier. It may also perform the allocation of the IP address. This component is installed in the backbone by the IAP to control the resource allocated per ISP.

TACACS

Terminal Access Controller Access Control System (TACACS) are protocols that allow a NAS to off load the user administration to a central server. There are now three versions of an authentication protocol that people commonly refer to as TACACS. The first is the ordinary TACACS, the second is an extension of the first, commonly called Extended TACACS or XTACACS, and the third one is TACACS+, which is not compatible with TACACS or XTACACS.

2.1.2.2 RADIUS proxy and RADIUS server

The RADIUS proxy interfaces with the RADIUS server during the roaming request for authentication. A detailed description can be found in Section 2.1.3, "Roaming between ISPs" on page 43.

2.1.2.3 Short Messaging Service Center (SMSC)

SMSC interfaces with the ISP platform for SMS messaging. SMSC is the interface between the GSM network and the ISP's IP backbone.

SMS was created as part of the GSM phase 1 standard. Historically, network operators needed to purchase a first generation SMS Center as part of the network commissioning plan. The initial SMS Center could be simply a voice mail platform module or, alternatively, a standalone SMS Center.

Now SMS Mobile Terminate Services are often offered along with voice mail notifications, which account for the vast majority of SMS traffic on the network (typically over three-quarters).

Moreover, the network operator now launches SMS Mobile Originate to give customers true two-way SMS capability. It has been observed that adding a SMS Mobile Originate typically leads to a 25 percent increase in overall SMS volumes being handled. Additionally, wireless Internet/mobile e-mail service can be provided, typically with the customer's mobile number becoming part of the e-mail address. They are allocated as part of the service.

The next quantum leap in SMS traffic volumes is caused by the introduction of SMS for prepayment customers. These customers pay for their cellular airtime as they go, rather than having contracts.

Each short message is up to 160 characters in length when Latin alphabets are used, and 70 characters in length when non-Latin alphabets, such as Arabic and Chinese, are used. Services usually start with mainstream content, such as news, travel, weather and sports.

Practically, SMS is a store and forward service. Short messages are not sent directly from sender to recipient, but always via an SMSC instead. Each mobile telephone network that supports SMS has one or more messaging centers to handle and manage the short messages. SMS requests are sent from mobiles to the ISP platform through the SMSC. The SMSC can create and forward the request with IP packets or generate an HTTP request. Replies or SMS notifications are sent through the SMSC to the mobile. Figure 12 on page 42 gives an overview of SMS.

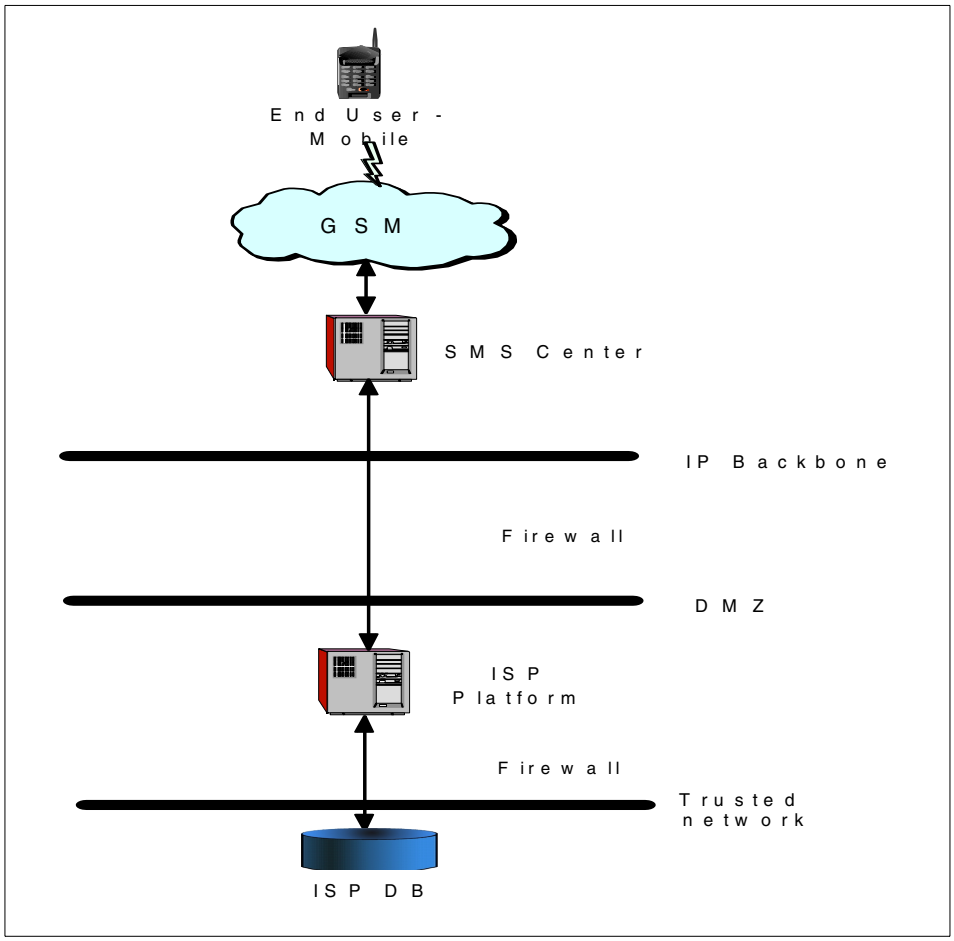


Figure 12. SMS messaging

2.1.2.4 GGSN and authentication server

GGSN interfaces with the authentication server during the connection creation from the GPRS network and the authentication. GGNS is the interface between the GPRS network and the ISP's IP backbone.

2.1.2.5 Proxy server

A proxy server interfaces with the authentication server when a request comes from the Internet. We will address this case in more detail in Section 2.2, "Security zone" on page 57.

Note that a proxy server primarily handles requests when a connection is created between an ISP user and the platform services or the Internet.

2.1.3 Roaming between ISPs

A subscriber usual has customer-vendor relationships with only one ISP. Nevertheless, there are examples where roaming capability is provided to subscribers. These include ISP confederations and ISP-provided corporate network access support. In these cases, we distinguish two types of ISP:

- Home ISP is the ISP with whom the user maintains an account relationship.
- Local ISP is the ISP who the user calls in order to get access.

Where roaming is implemented, the local ISP may be different from the home ISP.

The ISP confederations and ISP-provided corporate network access support are comprised of:

- *Regional ISPs* operating within a particular state or region, looking to combine their efforts with other regional providers to offer service over a wider area.
- *National ISPs* wishing to combine their operations with one or more ISPs in another nation to offer more comprehensive service in a group of countries or in a continent.
- *Businesses* desiring to offer their employees a comprehensive package of access services on a global basis. Those services may include Internet access as well as secure access to corporate intranets via a VPN.

Shared use network is an IP dial-up network whose use is shared by two or more organizations. Shared use networks typically implement distributed authentication and accounting in order to facilitate the relationship among the sharing parties. Since these facilities are also required for the implementation of roaming, implementation of shared use is frequently a first step toward the development of roaming capabilities. In fact, one of the ways a provider may offer roaming services is to conclude shared use agreements with multiple networks. However, to date, the ability to accomplish this has been hampered by the lack of interoperability among shared use implementations.

The requirements for successful roaming among an arbitrary set of ISPs include convergent methods to share dial-in numbers, connection management, authentication, NAS configuration and authorization, address assignment and routing, security, and accounting. The dial-in numbers

sharing solution is normally Web-based, and connection management between the subscriber and the ISP is done with Point-to-Point Protocol (PPP) or Serial Line Internet Protocol (SLIP). The other requirements affect the network infrastructure and the configuration.

2.1.3.1 Authorization: login to NAS

When a user wants to access the ISP platform, he needs to access the ISP LAN servers first. Dial-up and dedicated circuits access are the two ways of providing this remote connection.

The dial-up connections are made available through conventional telephone line and modems. The NAS is logically composed of a RAS and a RADIUS client. The RAS generally contains one LAN interface attached to the hub and many serial ports where the modems are connected. The first function of the RAS is to capture the authentication information from the client and then ask the authentication server (RADIUS client to RADIUS server) for approval. Once the authorization is approved, the protocol switches to PPP and the RAS gives an IP address to the client. The IP address is based on an user name and port or a pool of addresses. At that time, the client is part of the ISP LAN, switches to IP, and can request access to an application or to the Internet.

When the user accesses a local NAS, it provides its user ID either as "username" or "username@domain." The NAS will pass the user ID and password to the authentication server which supports RADIUS, TACACS, or both. If the "username" notation is used, the authentication server will assume that the user is a local user and will handle local authentication accordingly. If "username@domain" is used, the authentication server will process it as a roaming request. This is an example of how a proxy RADIUS works, as defined in Section 2.1.2.1, “NAS and authentication servers” on page 39.

When the authentication server handles a request from a roaming user, it will first check the cache to see if the user information is already stored there. If there is a cache hit, the authentication server will do the local authentication accordingly. If it does not find user information in its cache, it will act as a proxy, forwarding the authentication request to the home authentication server. When the home authentication server responds, the local server will forward the response to the NAS.

Caching is used to avoid frequent proxying of requests and responses between the local authentication server and the home authentication server. When the home authentication server sends back a valid authentication response, the local authentication server will cache the user information for

some period. The next time the user authenticates directly against the home authentication server, the home authentication server will send a request to the local server or servers to clear the user's information from the cache.

A roaming access topology example is shown in Figure 13.

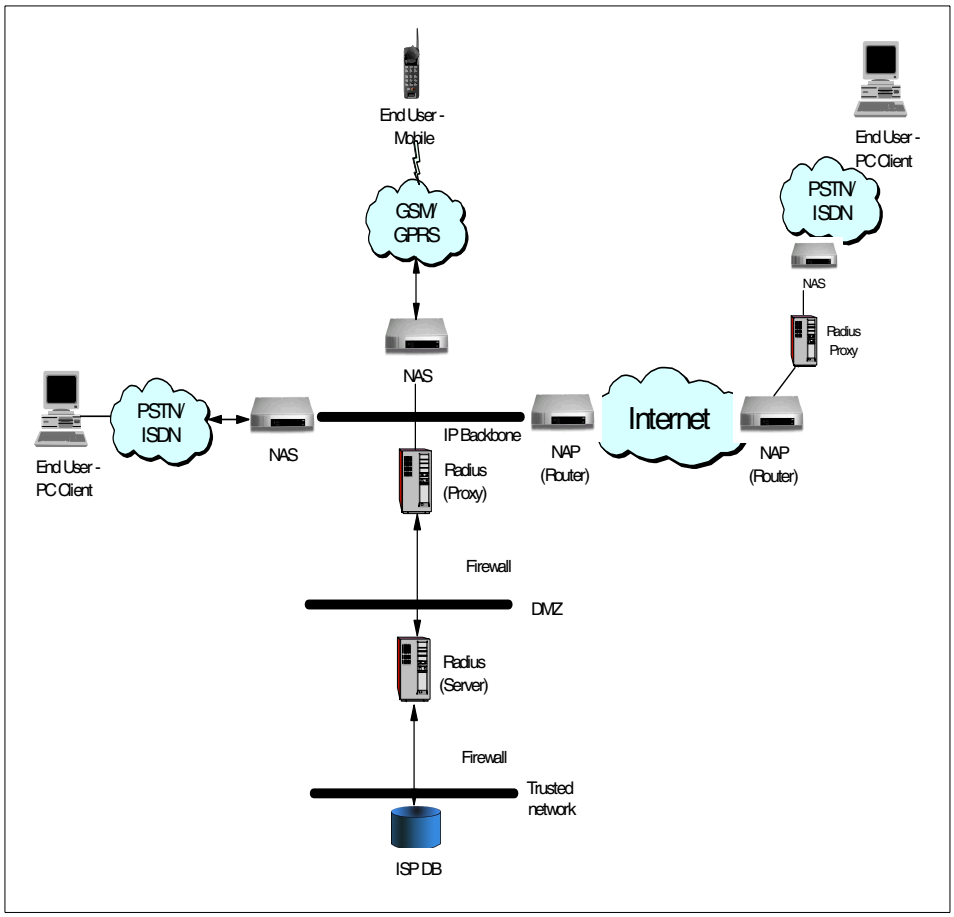


Figure 13. Roaming access with RADIUS based authentication

2.1.3.2 NAS Configuration and authorization

Although the attributes returned by the home authentication server may make sense to home NAS devices, the local NAS may be configured differently, or may be from a different vendor. As a result, it may be necessary for the local authentication server to edit the attribute set returned by the home authentication server, in order to provide the local NAS with the appropriate configuration information.

2.1.3.3 Address assignment and routing

Usually all addresses are assigned dynamically from within the address pool of the local ISP. Static addresses and routed LAN connections have been implemented in some software.

2.1.3.4 Security

Encryption is used between the local authentication server and the home authentication server. Public or Private key encryption is supported. IP tunneling and token card support is under consideration.

2.1.3.5 Accounting

Accounting transactions are sent to the home ISP when it is received from the NAS. This is intended to allow ISPs to update users credit limit information on a real-time basis. This assumes that this capability is supported by ISPs billing and accounting systems. Settlement is performed monthly.

2.1.4 Domain Name Service

In a hierarchical network, some hosts designated as name servers resolve names into Internet addresses for other hosts. There are several types of Domain Name Server (DNS).

A *primary* name server loads its data from a file. A *secondary* name server receives information from a primary name server. A third type of name server is a Caching Only Server (COS). Multiple DNS are required for redundancy. Redundancy is built into the software.

Every server and network component has one or more IP addresses and names. This information is updated on the DNS.

2.1.4.1 Domain name resolution

The main function of DNS is to translate names into IP addresses and vice-versa. The DNS is a distributed database. Specific database segments are locally controlled, yet data in each segment is available across the entire network through a client-server scheme. Name servers contain information about some segments of the database and make it available to clients, called resolvers.

The structure of the DNS database is that of an inverted tree, with the root at the top. Each node in the tree represents a partition of the overall database, or domain. Each domain can be further divided into partitions, called subdomains. A domain has a domain name, which identifies its position in the database, like host1.ibm.com. Each domain name is associated with its IP

address, like 123.45.78.101 for host1.ibm.com. The structure of a DNS database is shown in Figure 14.

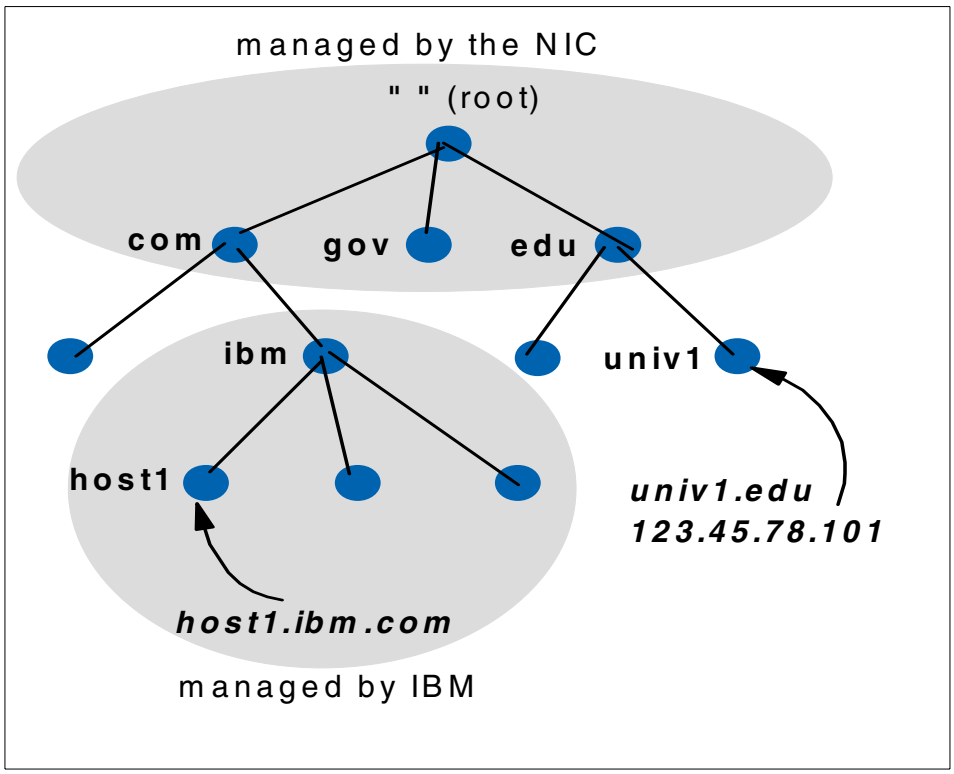


Figure 14. Structure of a DNS database

The domain name space is divided into zones which are administered by different organizations. The set of name servers which own a zone are said to have authority for that zone and are called authoritative. Authoritative name servers can be either primary or secondary masters. A primary master owns the data for the zones it is authoritative for. A secondary master gets its zone data from the primary master. Secondary masters provide the redundancy required to ensure the reliability of the DNS.

2.1.4.2 Resolvers

Resolvers are the clients that access name servers. They query the name server, interpret its response and return the information to the program that requested it. Resolvers vary in their level of sophistication. The most typical implementation is the stub resolver, which is quite simple and puts most of the burden of the resolution process on the name server. A full resolver, on

the other hand, has more functions and can off-load processing at the name server by performing tasks such as caching responses for future use.

2.1.4.3 DNS resolution process

Name servers resolve names into IP addresses by retrieving data from the domain name space. They can access data for their own zones, and can also search through the domain name space to find data they are not authoritative for (this process is called resolution).

To find its way to any point in the domain name space, a name server can issue a query to the root name server, which can then provide the names and addresses of the name servers authoritative for the relevant top-level domain, like com or edu. The top-level name servers can then provide the list of name servers authoritative for the second-level domain, like ibm, and so on. Each name server queried returns information about how to get closer to the answer, or provides the answer itself. In order to help speed up successive queries, name servers cache the data based on the data's time to live, which is determined by the zone administrator. After the time to live expires, the name server must discard the cached data and get new data from the authoritative servers.

Queries can be recursive or iterative. Figure 15 on page 49 illustrates the resolution process that takes place when a stub resolver issues a recursive query to a local name server. Recursive queries place most of the burden of resolution on a single name server, which follows successive referrals from other name servers until it finds the answer that is looking for. In the case of an iterative query, on the other hand, a name server simply gives the best answer it already knows (from its local data and cache) back to the querier.

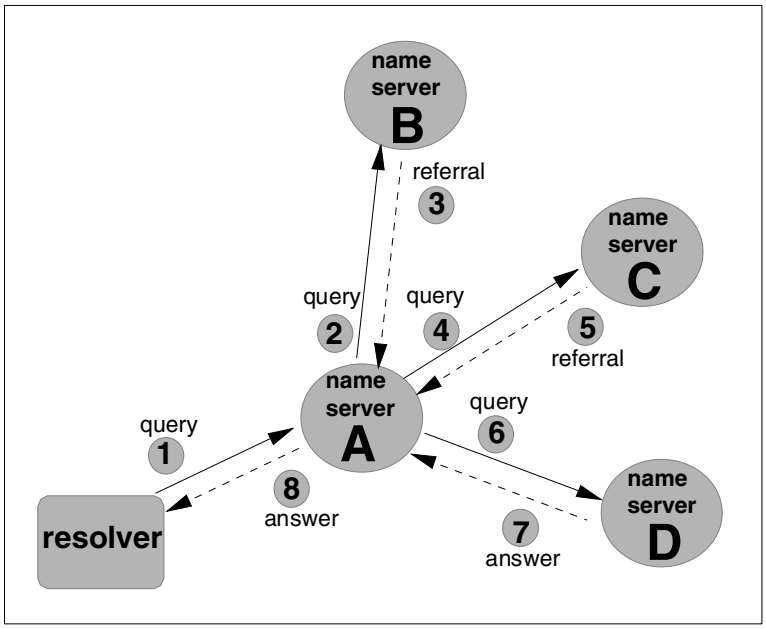


Figure 15. Resolution process

1. Name server **A** receives a query from the resolver.
2. **A** queries **B.**
3. **B** refers **A** to other name servers, including **C.**
4. **A** queries **C.**
5. **C** refers **A** to other name servers, including **D.**
6. **A** queries **D.**
7. **D** answers.
8. **A** returns answer to resolver.

Inverse queries allow a client to ask for the domain name of a host for which it only has the IP address. In order to enable this reverse translation, the IN-ADDR.ARPA domain was created. The IN-ADDR.ARPA domain Inverse queries allow a client to ask for the domain name of a host for which it only has the IP address. This domain uses addresses of hosts in reverse order to point to the hosts' domain names.

2.1.5 Network security

So far, we have given an overview of the network topology and have taken a closer look at the access network of the ISP platform, which handles the authentication of ISP subscribers.

Identifying subscribers is not close to being sufficient to guarantee security. This only provides security for subscribers as only they should have access to the (billable) services they are entitled to. Fully securing the customers and the ISP also implies that the ISP should provide:

- Data origin authentication

  Check that all data received is really originated by the sender.

- Data integrity

  Check that the data received has not been changed during transit.

- Data confidentiality

  Make sure data is encrypted in a safe way.

- Replay protection

  Make sure data is not a previously intercepted communication that is relayed by a third party.

Security applies on different layers. When applied to each TCP/IP layered protocol stack, protocols can contribute to protecting data in the four areas described above. This is shown on Figure 16 on page 51. Applications protection is handled in Section 2.2.2, “Security layers” on page 58.

We give here an overview of some of these techniques. Applying some of these techniques can create Virtual Private Networks (VPNs). VPNs can be a requirement for ISPs to implement. Therefore, we need to take a look at what a VPN is and when it is needed.

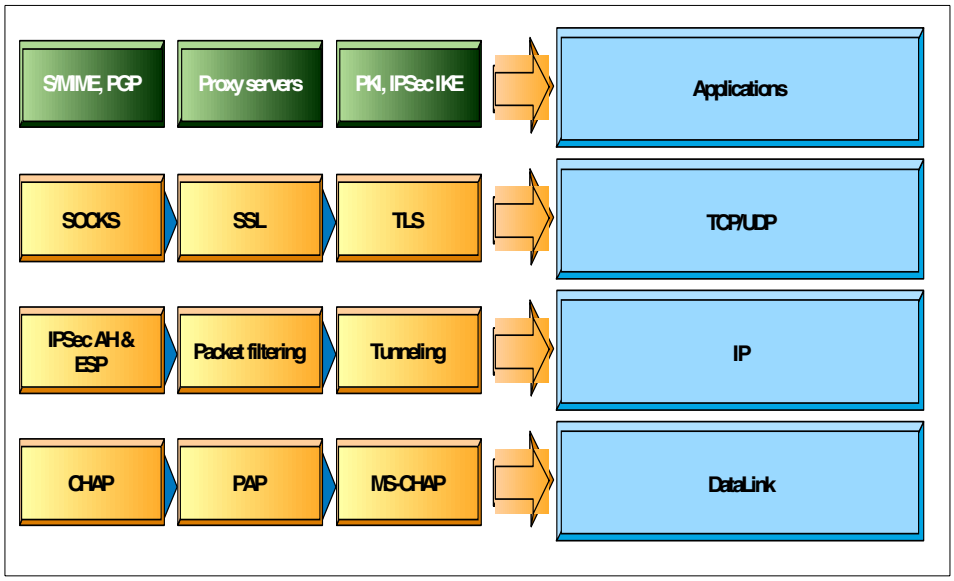


Figure 16. TCP/IP protocol stack and security protocols

Virtual Private Network (VPN)

We distinguished, in Section 1.2.5, “Packagers business model” on page 8, two types of customer: residential and business users. We have demonstrated that security was a key feature for businesses, and, in time, for residential users. Businesses are required to build VPNs to ensure security.

A VPN is an extension of an enterprise’s private intranet across a public network, such as the Internet, creating a secure private connection, essentially through a private tunnel. VPNs securely convey information across the Internet connecting remote users, branch offices, and business partners into an extended corporate network. ISPs offer, for example, cost-effective access to the Internet, via direct lines or local telephone numbers, enabling companies to eliminate their current, expensive leased lines, long-distance calls, and toll-free telephone numbers. In that sense, an ISP/ASP can become an extension of the company network.

We have outlined four areas where a VPN is likely to be required:

- Corporate remote access
- Branch office access
- Partners/suppliers access
- ASP

Corporate remote access

In Figure 17, we show how remote employees of a company may need to connect to their corporate network. In order to do so, the employee needs a POP either:

- Delivered by the employee's company.
- Delivered by a local ISP, as shown in Figure 17.
- Through a roaming access point.

If it is an ISP providing the POP, it will have to support one or many of the security protocols in its network access in order to create a VPN. The company itself is connected to the Internet through a Transport provider or a Packager, as defined in Section 1.2.4, "Segmentation" on page 4.

The company will need to choose the right level of service from the ISP. Depending on the service chosen from the list mentioned in Section 1.2.5, "Packagers business model" on page 8, ISPs may need to fully handle the VPN. Moreover, not one, but many distinct ISPs may need to collaborate in making the VPN work. For example, the employee could access a European ISP with his laptop and try to connect to an American ISP that handles his company.

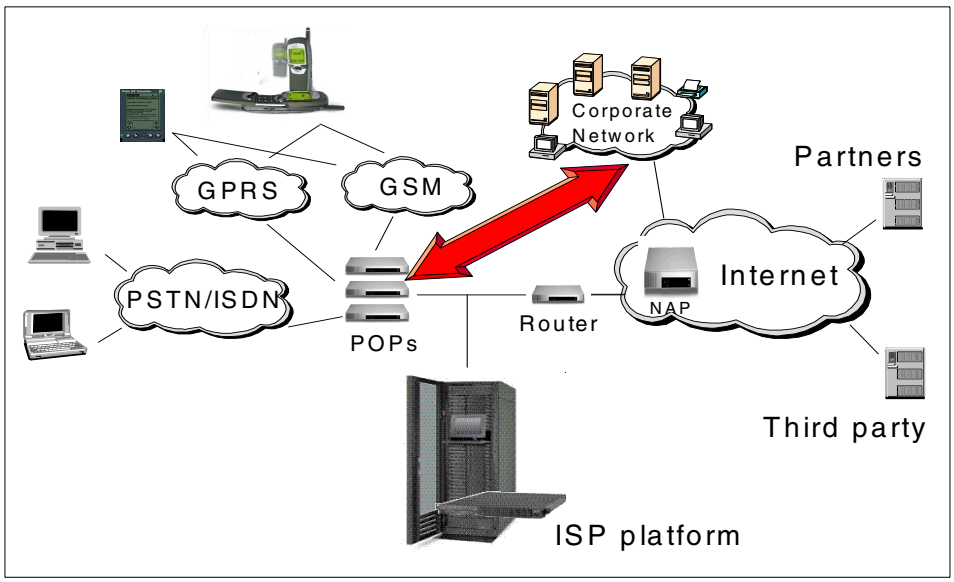


Figure 17. Corporate remote access model

Branch office VPN

In Figure 18, we show how a company maybe required to have a VPN in order to connect its branch offices.

As for the previous case, the company and, eventually, its branch offices are required to connect to the Internet through a transport provider or a packager. If the company chooses a basic Internet access, it will need to implement, from end to end, the VPN. Otherwise, the ISP may be required to handle the configuration of it. This is part of the managed communication environment offer; refer to Section 1.2.5, “Packagers business model” on page 8 for more details.

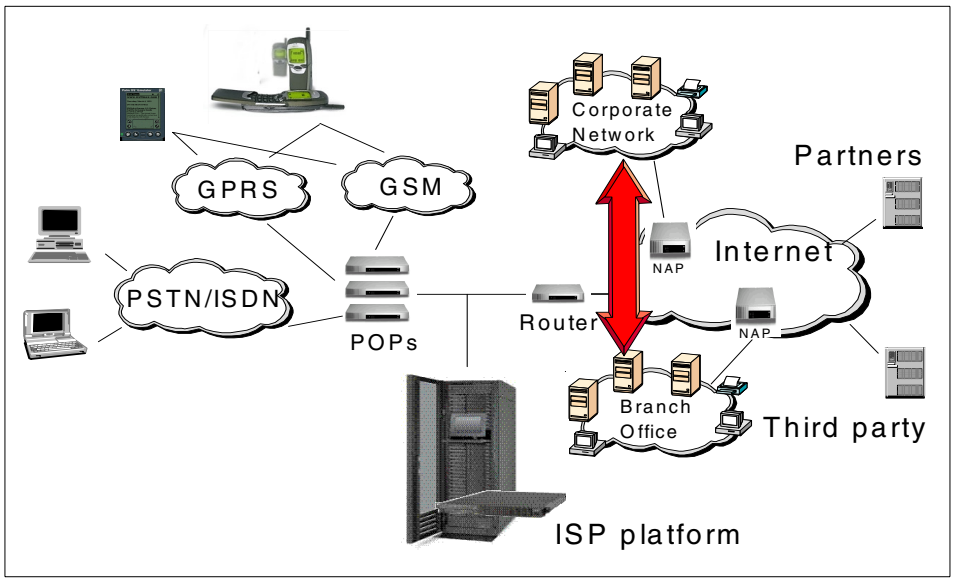


Figure 18. Branch office VPN model

Partners/suppliers VPN

On Figure 19 on page 54, we show how a company maybe required to have a VPN in order to connect to partners and suppliers. This is very similar to the previous case, where the same issues arise between partners and suppliers connected to the Internet through Transport providers or Packagers.

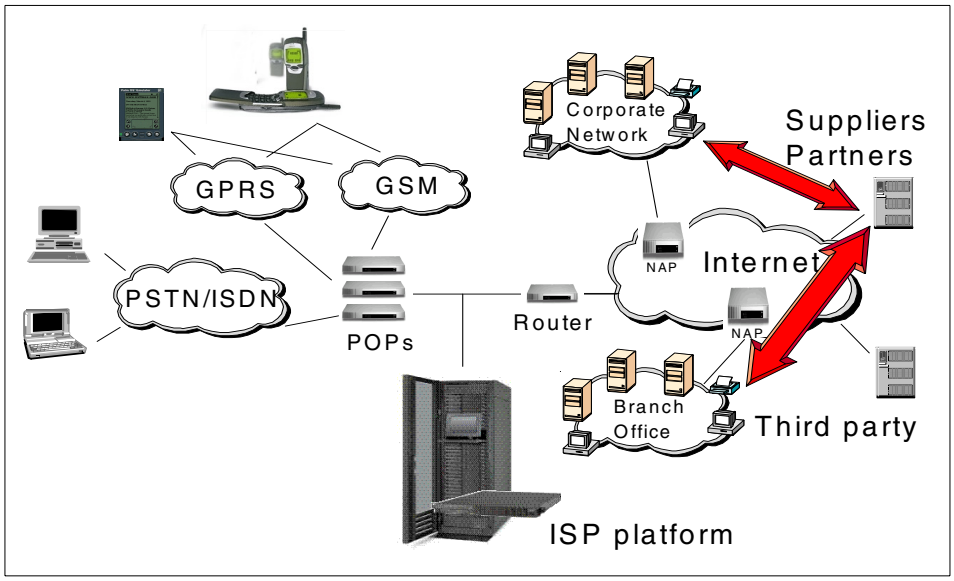


Figure 19. Partners/suppliers VPN model

ASP VPN

In Figure 20 on page 55, we show how a company or remote employees may require a VPN in order to connect to ISP/ASP services and applications.

The issue does not concern remote employees as long as they access directly through their home or local ISP. As soon as a roaming access is used, the traffic goes over the Internet, and a VPN may be required for increased security. Examples of services or applications that can be used from a remote location are collaborative tools, which can be centralized on the ISP/ASP.

Application layer security protocols are efficient means of protecting the data transmitted. Nevertheless, business critical information that is handled requires maximum security. A VPN is, in that case, mandatory.

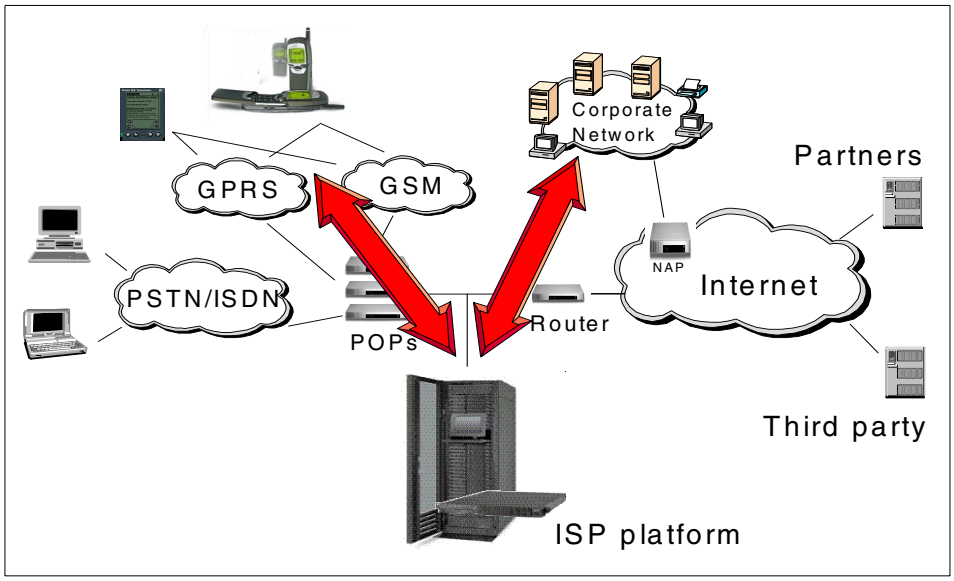


Figure 20. ASP VPN model

Implementing VPNs

Vendors who support a broad range of hardware and software VPN products provide the flexibility to meet these requirements. VPN solutions today run mainly in the IPv4 environment, but it is important that they have the capability of being upgraded to IPv6 to remain interoperable with VPN solutions. Perhaps equally critical is the ability to work with a vendor who understands the issues of deploying a VPN. The implementation of a successful VPN involves more than technology; the vendor's networking experience plays heavily into this equation.

We now take a look at the solutions that can be implemented. Two approaches can be taken to implement a VPN:

- Network layer-based with IPSec
- Data link layer-based with L2TP

IPSec-based VPN solution

Within the TCP/IP stack model, the network layer (IP) is the lowest layer that can provide end-to-end security. Network-layer security protocols provide blanket protection for all upper-layer application data carried in the payload of an IP datagram without requiring a user to modify the applications.

The solutions are based on the IP Security Architecture (IPSec) open framework, as defined by the IPSec Working Group. IPSec is called a framework because it provides a stable, long lasting base for providing network layer security. It can accommodate today's cryptographic algorithms, and can also accommodate newer, more powerful algorithms as they become available. IPv6 implementations are required to support IPSec, and IPv4 implementations are strongly recommended to do so.

The principal IPSec protocols are:

- *IP Authentication Header (AH)* provides data origin authentication, data integrity, and replay protection.
- *IP Encapsulating Security Payload (ESP)* provides data confidentiality, data origin authentication, data integrity, and replay protection.
- *Internet Security Association and Key Management Protocol (ISAKMP)* is now referred to as IKE, and provides a method for automatically setting up security associations and managing their cryptographic keys.

The IP Authentication Header (AH) provides per-packet connections, integrity, and data origin authentication for IP datagrams, and also offers protection against replay:

- Data integrity is assured by the checksum generated by a message authentication code.
- Data origin authentication is assured by including a secret shared key in the data to be authenticated.
- Replay protection is provided by use of a sequence number field within the AH header.

The IP Encapsulating Security Payload (ESP) provides data confidentiality (encryption), connectionless (that is, per-packet) integrity, data origin authentication, and protection against replay. ESP always provides data confidentiality, and can also optionally provide data origin authentication, data integrity checking, and replay protection. Comparing ESP to AH, one sees that only ESP provides encryption, while either solution can provide authentication, integrity checking, and replay protection. When ESP is used to provide authentication functions, it uses the same algorithms used by the AH protocol.

Layer 2-Based VPN solutions

Layer 2 Tunnel Protocol (L2TP) is a tunneling protocol. L2TP extends the span of a PPP connection; instead of beginning at the remote host and ending at a local ISP's point of presence (PoP), the virtual PPP link now

extends from the remote host all the way back to the corporate gateway or a remote ISP platform. In effect, the remote host appears to be on the same subnet as the corporate gateway or the Internet platform.

Since the host and the gateway share the same PPP connection, they can take advantage of a PPP's ability to transport protocols other than just IP. For example, L2TP tunnels can be used to support remote LAN access as well as remote IP access.

Although L2TP provides cost-effective access, multiprotocol transport, and remote LAN access, it does not provide cryptographically robust security features. This is why the IPSec protocols are usually used to protect the data that flows through an L2TP tunnel.

2.2 Security zone

Security applies to all components of the architecture. The first layer of security is the network security, which guarantees privacy of communication over the network. Network security is described in Section 2.1.5, "Network security" on page 50. A second layer of security is the authentication of registered users connecting to the platform via Telco infrastructure. Specific protocols are used, such as RADIUS or TACACS. This is part of the access network described in Section 2.1.2, "Access network" on page 38.

Security also applies when authenticating registered users when they access the platform from the Internet and, more generally, on each request they make. This is the authentication process we describe in this chapter, which must not be mistaken with the authentication on first connection from Telco networks described in Section 2.1.2, "Access network" on page 38.

We then describe the security principles and the components of this security that can be implemented on the platform for different levels of security enforcement.

2.2.1 Authentication

Authentication is the process of verifying the identity of a user. This can be done in a variety of ways, but they all rely on a single concept. The concept is that a user must be who they claim to be, if they can prove they are in possession of something, known as the secret, that is held by only by them.

Authentication is the process through which a subscriber is identified when he tries to dial-in or log-in to the ISP. The user usually provides a username and password to get access to his home page. If the subscriber is dialing in, this

input is given at connection time. If the user had other online access, he can usually log-in through a link on the ISP main portal page. Mobile device user authentication is, as for Web enrollment, not done through the same process. The calling number (MSISDN) of the mobile device can be used to authenticate the subscriber without any username and password input.

Authentication mechanisms are described in Section 2.3.3.3, "Personalization" on page 68. Personalization is an example of service that uses intensively authentication of a user. Another example is described in Section 2.7.2, "Service management" on page 89, where single sign-on functionality is explained.

2.2.2 Security layers

This chapter focuses on those elements of security that are required to provide secure access to the ISP infrastructure; more specifically, to data and applications.

Here are three bases for architecture security:

- Front-end servers, whose purpose is to deliver services to subscribers. Front-end servers should be located in a subnetwork that all subscribers can reach.
- Back-end servers, which house subscribers' profiles and all business critical data, should be located in a different subnetwork that can be accessed only from the front end servers.
- From a security perspective, the purpose of layering is that any intrusion on a given server will limit the harm to the service hosted on this server. Assuming that several servers are used to provide the same service, the service will remain available, but potentially in degraded mode.

2.2.2.1 Access to ISP infrastructure

The traditional way to secure a network is to tightly control the traffic that is allowed in. Usually this involves filtering of the incoming traffic. This kind of security is known as boundary security. We need to protect against malicious traffic, but also need to allow incoming requests through, so that Web and application servers can be reached. ISPs want people from the outside to access their internal resources in a controlled way. These conflicting requirements define a compromise on what restrictions to apply on the services delivered by the ISP.

Access to business processes

Once the decision has been made to allow incoming requests to pass through the boundary security, the next issue is to decide which of those requests are

valid and should be processed, and which are illegal and should be rejected. The pre-requisite of making this decision is to know who made the request. This process is known as authentication. Once the user is known, the request they have made can be checked against what they are allowed to request, and then a decision can be made. This process is known as authorization.

Elements of security

Securing any network attached to the Internet is complex, and the problem is made significantly harder when it is necessary to allow, initially anonymous, users to access an ISP's internal systems.

Top down Security

In order to make the security of a network manageable, it is useful to consider the components of a security solution as if they were layers in a filtration device. Entering the filter is data traffic that contains any number of potential contaminants. We need to ensure that all of these contaminants are removed so that only uncontaminated legitimate requests reach critical data and business processes. To accomplish this, there needs to be at least one layer in the filter that handles each different type of contaminant. Authorization and the pre-requisite authentication should also be considered as a layer in the filter. Together, they provide the ability to limit access to the protected data and business processes based on an authorization policy.

2.2.2.2 Components for security implementation

Firewalls are essential components to create a secure ISP platform. We will use one firewall to create a region of the network between the Internet and the internal network. This region is referred to as the Demilitarized Zone or DMZ, and contains the servers that users on the Internet will access.

In addition to the firewall that guards the connection to the Internet, there will be other firewalls that define boundaries between areas of the ISP platform with different security needs.

Other firewalls may be placed between any third application network on one side and the DMZ and/or the trusted network on the other side, depending upon the allowed communications between these subnetworks.

Another addition to this situation will be a third firewall that defines the trusted network, which has very restricted access, and contains the servers that need to be protected from internal as well as external users. This is an important consideration since a significant percentage of attacks to networks originate from the internal network rather than from the Internet.

Firewalls

Often used as the first line of defense for a network, a firewall has two main functions. The first is to protect against attacks based on network and transport layer protocol exploitation, such as address spoofing or TCP SYN attacks that could cause an unprotected host to crash or have its service degraded. The second is to filter traffic so that only certain protocols are allowed into the network and that those protocols are only passed to the appropriate servers that deal with them. To ensure that these functions are performed effectively and reliably, the firewall itself must be protected against unauthorized access and damage. This protection is often referred to as hardening. A firewall must be capable of detecting and dropping any illegal or malformed packets without damage to itself. In order to prevent a firewall from being compromised, it is common practice to deny any administrator access from the unsecured side. This means that even if a hacker were able to get administrators access details, he would have to gain physical access to the internal network in order to use them.

Traditional proxy

A traditional proxy is used to allow internal users to gain access to external servers without having to give everyone a direct access through the firewall. The internal users all connect to the proxy and this connects through the firewall to the destination. This allows the firewall to be more secure, provides control and auditing of the use of the Internet, and hides the user's IP addresses from the external servers.

Reverse proxies

One of the functions a firewall performs is to ensure that each protocol permitted into the network is only allowed to reach the appropriate hosts. This is an important function, but the next question is whether or not these hosts are robust and secure. The servers (or daemons) for applications, such as HTTP, TELNET, FTP, DNS, and MAIL are not always written with security in mind. They may work correctly in normal operation, but they may be vulnerable to attack if the protocol rules they expect clients to adhere to are broken. They may also lack the capability to do the checking required to prevent abuse of their services. Rather than trying to update all of the daemons of each type in the internal network, it is often more secure, and easier to administer and audit, to set up a proxy for each type of service.

Proxies are written so that they can withstand requests that do not adhere to the rules of the protocol they use. They also contain code to ensure that their services are not open to abuse. The proxy acts as an intermediary; the client connects to the proxy and, after the request is validated, the proxy connects to the server. At no point is there any direct communication between the client and the server, so the server is protected. This arrangement also has the

advantage that an external user never sees the IP address of the server (since they only ever communicate with the proxy). This makes it much harder for a hacker to mount an attack on the server, even if they somehow get direct access to the network. These proxies are called reverse proxies.

The reverse proxies discussed previously work in the opposite direction from these original proxies, allowing external users to access internal servers. Reverse proxy functionality will be provided by the firewall.

Content filtering

Even with a firewall and application proxies in place, it is still possible for an attacker to send malicious code into the network. One of the easiest ways for them to do this is to trick a valid internal user into unwittingly downloading code from the Internet.

Executable code can be received in a variety of ways. The most common ways are downloading it from a Web or FTP site, or receiving it as an e-mail attachment. If the malicious code is executed on an internal user's machine, it potentially has access to all the internal systems that user can access. It can then damage them or send details about them to a malicious user on the Internet.

To alleviate this threat, files downloaded from the Internet must be checked before they are executed. This could be done on every client machine, but this puts the responsibility in the hands of the user, and they may not have the technical knowledge or the inclination to check every file.

A more thorough method is to automatically pass all downloaded files through a content filter before the user receives them. This process is transparent to the user and protects their machine and the rest of the network. Content filtering can only work if all user traffic passes through the filter. As a result, content filters usually work in the same way as a proxy; all users send their requests to the filter and it passes them on. The resulting response is passed back via the filter, which can examine any files it finds before passing them back to the user. If a given protocol, for example FTP, requires both a proxy and a content filter, then that traffic is passed through each in turn; the process is known as chaining.

Public key infrastructure (PKI)

PKI technology is emerging as the preferred trust mechanism for networked business. A key component of PKI is the certification authority that provides digital credentials and security features, such as message integrity, data privacy, signature verification, and user authentication. Certificates, with associated roles and privileges and registries to manage certificates, and key

pairs, that support digital signing and encryption, provide the quantum step-up needed to reconcile the Internet/security compromise.

2.3 Split between front end, back end and ISP services

We will describe, in this chapter, the motivations for splitting the infrastructure into two separate zones. In addition to the security reasons explained earlier, this configuration is also aimed at allowing scalability, high availability, and flexibility.

Applications that run on the front end or the back end follow a component model, which is a multi-tier architecture. Each tier is deployed forming a generic seven layer infrastructure. We describe this model in this chapter.

Finally, we will take a look at the applications or services that are the root of services offered to customers by ISPs. These services are deployed according to the seven layer infrastructure.

2.3.1 Seven layer infrastructure

To describe the split between the front end zone and the back end zone, we introduce a component layering model. It is a multi-tier application model that explains how the architecture is and should be split into a front and back end configuration. The distinct layers are:

- Presentation layer

  The process enables structuring of information for client devices that use distinct formats, such as HTML or WML.

- Transformation layer

  The process implies transforming content from one format into another, such as with XML or JSP, in order to manage only one type of format for all other processes in the deeper layers.

- Abstraction layer

  Generic and specific interfaces are provided to build up customized functions on top of the ISP platform.

- Application layer

  The business logic is implemented using components such as session beans or access beans.

- Persistence layer

While supporting the applications, the components interface to data and map the data model to the object model.

- Data layer

  These are the tools and means that allow access to content data.

These layers define, in an ordered way, the processing of incoming and outgoing requests on the platform. The technologies used to support this layering are critical. There are many choices of technology to make, but there are also many implementations of these standards. The choices of middleware products to support this model may vary depending on performance, completeness, or flexibility criteria. The layering can also be applied onto the hardware configuration in order to gain the benefits of all those criteria.

The mapping of this model to a front and back end zones model is not trivial. The abstraction layer can apply to front end services as well as back end applications. The application layer can be implemented in either zones. Products are often implemented to allow, or not, both implementations. One major factor that is beyond the scope of this book is the deployment of a firewall between the front and back end of the system. Such an implementation, which we encourage deploying each time, can have an impact on the applications themselves, or applications can dictate the type of configuration possible for the firewall.

Outside the scope of such consideration, it is recognized that business applications should run in the back end zone, and such configuration should be employed each time it is possible.

The first two layers of the model do apply to the front end zone, as described in Section 2.4, “Front end zone” on page 69, and the four last ones will be described in the back end zone in Section 2.5, “Back end zone” on page 76.

2.3.2 Benefits to non functional requirements

As shown on Section 1.3.2, “Architectural evaluation criteria” on page 20, nonfunctional requirements, such as performance, scalability, high availability, flexibility, and security, benefit from the split into two zones of the architecture.

The physical architecture provides a general structure for the ISP platform. The access network is described in Section 2.1.2, “Access network” on page 38. It provides channels to send and receive data. The access network provides the entry flow of the ISP platform. This entry flow enables communication by using different communication protocols between nodes

that are implemented in a distributed way. We have already described the protocols that are generally used for communication implementation.

Front end zone
The front end zone only processes incoming requests. It does not hold any user information. No specific techniques are required in this zone to share or duplicate data. The front end layer is basically a set of servers that runs fully independent with virtually no coupling at all. These servers are usually called protocol nodes, as they are communication interfaces. Protocol nodes dispatch service requests to data access nodes.

The front end zone is on the Demilitarized Zone (DMZ), which is separating the Internet from the Secure Zone. Firewalls should be used between each zone.

Back end zone
The back end zone stores all customer and business critical data. Servers running on the back end zone are called data access nodes. These nodes must be implemented on a trusted network or secure zone. This is what the separation brings in for securing the environment. Moreover, by clearly separating the front end and back end functions, it is possible to apply the most appropriate techniques to achieve scalability and high-availability on the ISP platform. A more detailed description of security implementations can be found in Section 2.2.2, “Security layers” on page 58. The layering of the front end and the back end is shown in Figure 21 on page 65.

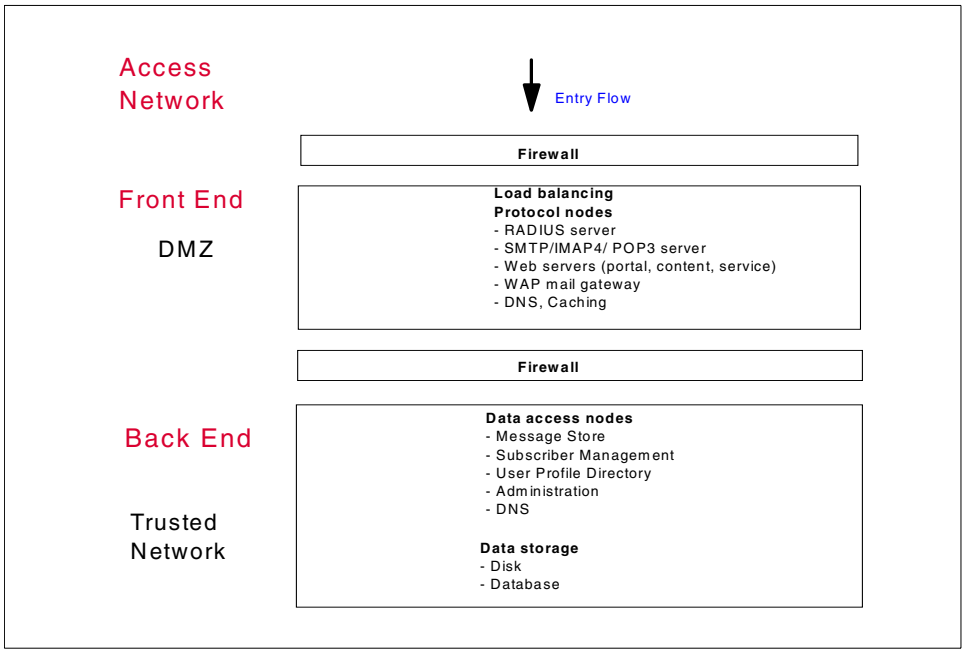


Figure 21. ISP platform layering

This architecture has several advantages:

- It is very scalable. The ISP has good granularity on the front end. A number of machines can be added or removed to adapt in accordance to the users' behavior. The back end is less granular. It is preferable because services in the back end do not change often. Such operations should seldom be achieved. Powerful machines are often chosen on the back end due to the nature of the operations and to withstand rapid volume growth without intervention.
- It is very flexible in regards to implementing security solutions. Indeed, it is possible to put security components, such as routers or firewalls, in front of the platform or between the front end and the back end. It is possible to define a DMZ composed of the front end machine (containing no sensible data) and a secure Local Area Network (LAN) composed of the back end machine. With different firewalls to protect each zone, security implementation becomes easier, firewall configuration is simpler, and the risks of human error in setting up these configurations is dramatically lowered.
- Elements of the platform can be distributed. Thus, some elements of the front end can be used to access servers located on other sites.

The benefits of such an architecture are demonstrated in Chapter 5, “How do ISP components fit in an RS/6000 SP?” on page 197.

2.3.3 ISP services

The ISP services described here are the business services that are built on top of the seven layer infrastructure. These services are built using the provided middleware technologies and the layering decomposition described in Section 2.3.1, “Seven layer infrastructure” on page 62. Using layers enables to fully scale the solution up to the maximum ability offered by the framework. By distributing components, their development is also greatly improved, as a single business function can be developed by many people in parallel. This is a strong feature of the seven layer infrastructure that serves, at its best, those services.

An ISP requires some basic functions of subscriber management to efficiently build up its subscriber base. These services are described here.

We will take a look at the enrollment processes considering all major use cases an ISP may need to implement, depending on its business model and existing infrastructure. We will then look at a service that allows subscribers to modify their personal information online. We will show what the benefits are for enabling this.

Finally, we will introduce the personalization service, which is a very important feature that is necessary for an ISP to achieve stickiness of users. We will then look at the benefits of such a service and give a brief overview of the mechanisms behind the feature.

2.3.3.1 Enrollment

New subscribers can be enrolled through:

- Customer care
- Web enrollment (subscriber self-enrollment)
  - From dial-up networking
  - From the Internet
- Mobile device enrollment specifics
- Specific bulk enrollment applications

Customer care enrollment is done by a Customer Service Representative (CSR). The customer calls the CSR, who then uses an internal tool to register the new subscriber over the phone. The internal tool used can either be a

customer care engine directly connected to the ISP platform (as a tool delivered within the ISP solution) or a legacy application tool.

Web enrollment is the application used by new subscribers to register themselves with the ISP. This can be performed through dial-up networking, as well as through Web access. Dial-up networking can be used by subscribers who come to the ISP using a kit, such as a branded CD-ROM, that dials into the ISP and initially logs on with limited access. This gives the new subscriber access only to the enrollment application. Web access allows new subscribers who are already connected to the Internet to enroll in an ISP using their current Internet connection. For example, a subscriber of another online service could navigate to the ISP enrollment page and enroll that way. At the end of enrollment, information about the subscriber is inserted in the subscriber base. An example of enrollment flow is shown in Figure 22.

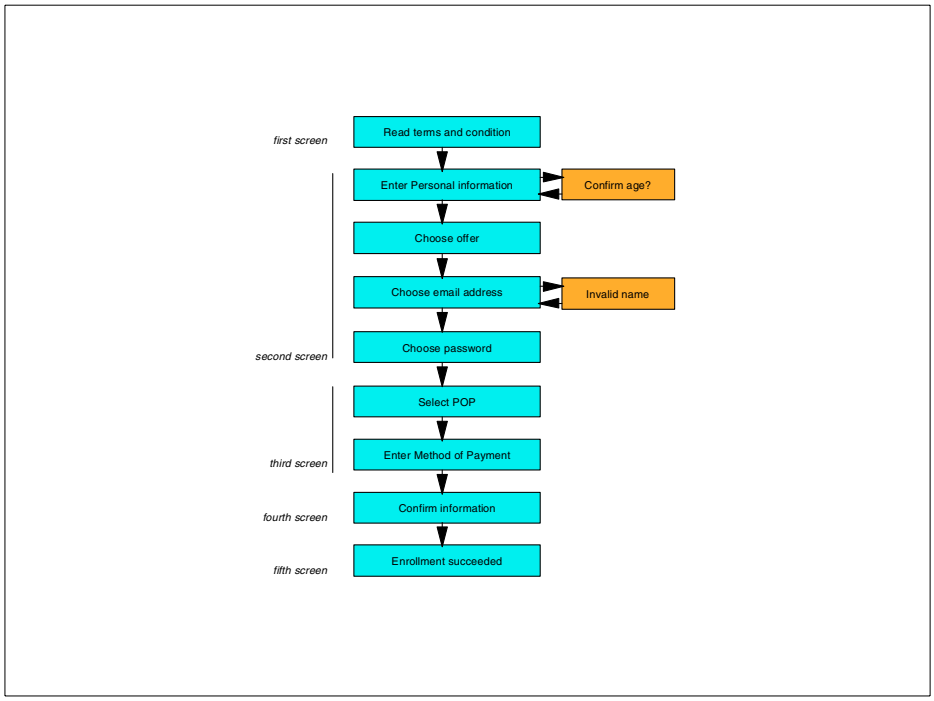


Figure 22. Enrollment navigation example

Mobile device enrollment is a specific case of Web enrollment. In most cases, entering any kind of information from a mobile device is very difficult. Mobile devices will be pre-enrolled in the ISP and will automatically get access to the ISP at first logon.

Bulk enrollment is how an existing set of customers that the ISP may have from other activities can be registered, in one operation, into the ISP subscriber base.

2.3.3.2 Customer self care

Customer self-care is the application that enables subscribers to modify their registration profile. This profile contains all personal and credit information recorded during enrollment. The usual operations a subscriber will want to do include:

- Change their password.
- Change their deal by subscribing to different offers or services.
- View their account status.
- Change method of payment.

Profiles are defined by the ISP, who can customize the amount of information to input. This is crucial to achieve personalization of the portal pages.

2.3.3.3 Personalization

Personalization allows each subscriber to have Web pages customized for their specific use. This is the most important business feature for an ISP. This process can be based on:

- The use of user-specific information to adapt the content of the pages to the interest of the user.
- The use of preferences that allow the user to modify, from his browser, the appearance and content of his pages.

User specific information can either be taken from the user profile stored in the subscriber base or come from user behavior analysis data that resulted from tracking the subscriber Web navigation preferences. (The information can come from both.) Pages can then be automatically tailored to contain links with content that may interest the user.

A prerequisite to achieving personalization is to identify who issued the HTTP request to the Web server. An authentication checker component can be used to identify the subscriber issuing the request. There are many ways of identifying the origin of a request. The choice between methods can be motivated both from technical and business criteria. Methods include:

- Check the IP address of the originator on the IP stack.
- Check the HTTP header of the request for the originator identity.
- Use a cookie downloaded on the client device.

The personalization component provides the framework that allows the ISP to manage the pages of their subscribers in a simple and ordered way. Additionally, preferences modifications enable the user to modify the look and feel of the pages. It can also let subscribers select, for example, their favorite links.

2.4 Front end zone

We have described, in the previous sections, the access zone and security zone of an e-business infrastructure. We have described the first two communications a registered user would accomplish when connecting from a Telco network, which are:

- Connection creation and authentication process
- DNS technology and function

We have also looked at considerations of security and how it can be implemented across the infrastructure. We will now look at the next communication that can occur on the platform, which involve services in the front end zone.

We will describe the most common types of request that can be made on the ISP platform. The types of requests depend upon the types of applications used by subscribers. Processing of the requests involve the presentation layer and the transformation layer, as described in Section 2.3.1, “Seven layer infrastructure” on page 62.

Front end servers are indeed protocol servers. We describe the physical components required for an efficient front end zone and the features of the implementation of such an architecture.

2.4.1 Presentation and transformation layer

The presentation layer is the component that enables a device to display the data it receives. This is an embedded component in the device. For World Wide Web (WWW) applications, for example, the component embedded can range from full Web browsers to micro browsers (on smaller devices).

The transformation layer handles the transformation of the data format and adapts it to the specific type of device that will receive the information. For example, data can be constructed in XML format and transformed in HTML or WML, depending on the target device.

2.4.1.1 Transport protocols

These layers are connected by different types of network running distinct suites of protocols. We describe here key protocols that are handled and why they are used.

Web protocols and TCP/IP

Web protocols are protocols at the application layer of the TCP/IP protocol suite. The TCP/IP protocol suite is structured in layers, as shown on Figure 16 on page 51. It is important to understand that the TCP/IP protocols are used on the basis of a reliable network. This basis is the datalink layer, which is the closest interface to the actual physical network. The other stacks are built on top of this datalink layer, and have particular functions:

- IP for routing
- TCP for fragmenting raw data and sequencing
- Application for data format and Web protocols

WAP

The Wireless Application Protocol (WAP) is a set of protocols targeted at wireless data communications. Because wireless network properties and features are different from traditional ones, such as PSTN, LAN, or WAN, TCP/IP is not usually used over these networks, even though it is possible to do so, for example, connecting laptops. Instead, WAP protocols are most likely used, because of low bandwidth, high latency, less reliable connections, and support devices that have small displays, limited memory and CPU power, and limited input capability.

The WAP protocols and architecture are designed to provide Web contents to mobile terminals, such as mobile phones and communicators. WAP is designed to support many terminal devices and different bearers.

The WAP protocols are depicted in Figure 23 on page 71. As for TCP/IP protocols, they are organized in layers.

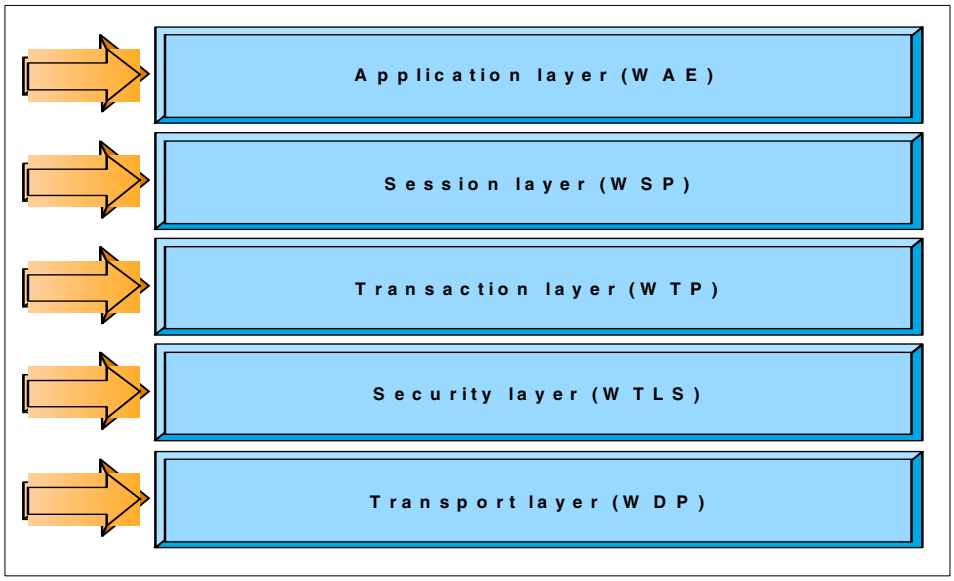


Figure 23. WAP protocol stack

HTTP

HyperText Transfer Protocol (HTTP) is the TCP-based protocol used by World Wide Web (WWW) applications, which enables the interaction between Web browsers and servers. HTTP is a stateless protocol designed to support hypermedia information transfer.

The HTTP protocol is based on a request/response paradigm. The client establishes a connection with and sends a request to the server. The server sends the response and then closes the connection. A feature of this protocol is the typing and negotiation of data representation, allowing systems to be built independently of the data being transferred.

HTTPS

HTTPS is a secure version of HTTP using Secure Socket Layer (SSL) as a transport-layer security mechanism that provides a secure pipe between the Web server and the Web client (browser).

WTP

The Wireless Transaction Protocol (WTP) is a light weight transaction-oriented protocol that is part of the WAP protocol, as shown in Figure 23 on page 71. It is used for implementation with thin clients, which are described in Section 2.1.1, “Network infrastructure” on page 31. WTP is similar to HTTP within the TCP/IP protocol suite, as it enables interaction

between Web browsers and servers. In fact, WTP translates into HTTP when it reaches the TCP/IP network, as shown on Figure 24. This is generally done by a gateway. WTP only operates over wireless datagram networks.

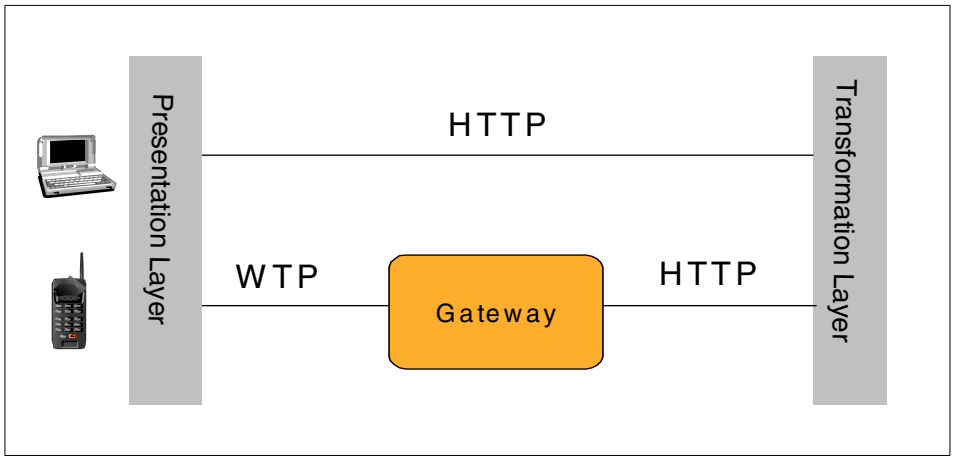


Figure 24. Transport protocols

2.4.1.2 Flow and content

The data transported over the networks are device specific. We give examples of these data protocols and how flows are handled with TCP/IP. We show how data is transported over the networks in Figure 25 on page 73.

HTML

Hyper Text Markup Language (HTML) is a markup language derived from Standard Generalized Markup Language (SGML). SGML is used for describing the format and the content of documents separately. A definition file is used to determine the format of the document. A document is generated as an instance of this definition file. HTML is a simplified version of SGML. The definition of the formats is static and the possibilities given are limited.

To overcome the limitations of HTML, other formats have been created, such as DHTML, VRML, and XML, which are not within the scope of this book.

WML

Wireless Markup Language (WML) is a lightweight markup language, similar to HTML, but optimized for use in mobile terminals.

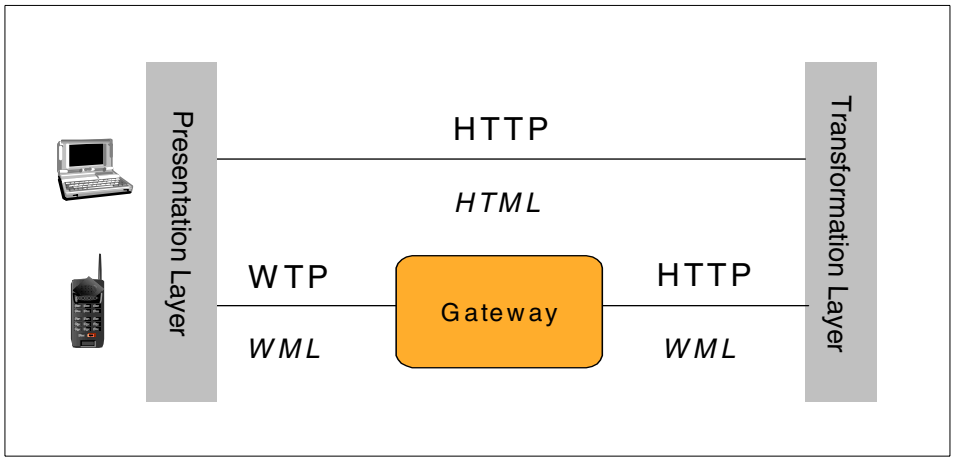


Figure 25. Data protocols

Client/server data flow with TCP/IP

Web protocols generally use the client/server model of interaction. Users invoke the client part of the application, which builds a request for a particular service and sends it to the server part of the application, using TCP or UDP as the transport vehicle. The server receives the request, performs the required service, and sends back the results as a reply.

Servers can usually deal with multiple requests at the same time. Some servers wait for requests at a well-known port so that their clients know to which channel (IP socket) they must direct their requests. The client uses an arbitrary port for its communication. Clients that wish to communicate with a server that does not use a well-known port must have another mechanism for finding out which port they must address their request to.

In Figure 26 on page 74, the HTTP and WTP requests coming on the ISP platform are represented. A Uniform Resource Locator (URL) request is sent from different devices. The URL represents a unique location, Internet wide. For example:

- An HTML page can be found that is stored on the ISP platform.
- An operation is requested to generate a page with dynamic content.

In the second case, a number of operations are required that may involve all the other infrastructure layers. Those components are represented and described in the next sections.

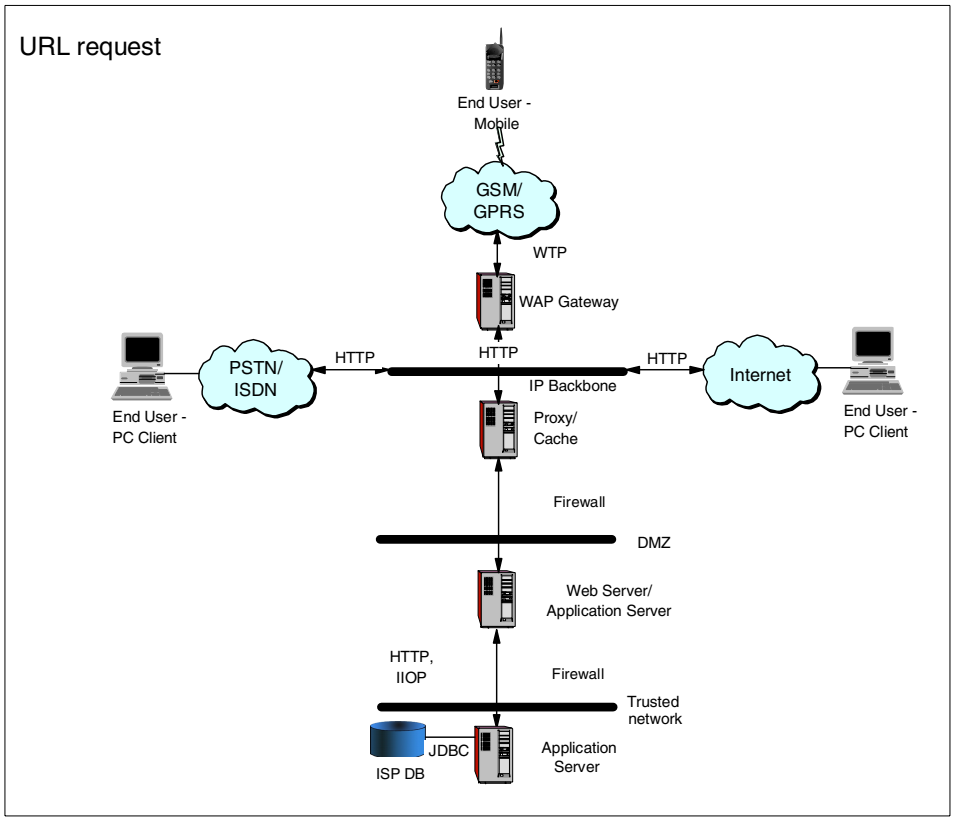


Figure 26. URL request from different networks

2.4.1.3 Transformation layer and transcoding

Transcoding is the process of transforming content from one format into another, including conversion between alternative screen sizes or window sizes and aspect ratios so that the content can be displayed on a wide and growing variety of devices. Both enterprise and Web content may be filtered, transformed, converted, or reformatted, to enable it to be universally accessed by a variety of devices, to exploit specific application requirements for content customization, and to enable personalization of general content. Moreover, this content may be delivered over a wide range of networks and, as a result, the network bandwidth and latency encountered will vary greatly.

2.4.2 Physical implementation features

The front end servers, which are responsible for receiving and managing the users' requests, are generally composed of the same type of node. The

advantage of this solution is that you can build a scalable solution that can cope with an increase in traffic on the platform. In fact, the same node is used for front end servers and can also be stored in a rack. The configuration of a node is very simple; the front end is based on clone machines that can be easily replaced, added for maintenance reasons, or for traffic increases.

There is not any user information on these nodes, so any node can manage user requests. This is an important point for the load balancing aspect and the high availability of the service. Only applications dealing with protocol services like SMTP, POP3, IMAP4, or HTTP are installed on these nodes. The front end machines are called protocol servers.

The key element to successfully build this type of architecture is the load balancing product. The choice of product determines the high availability and scalability level achieved for the platform. It is the unique entry point for all entry flow on the platform, and can be coupled with firewalls. Firewalls and the load balancing products must be doubled for high availability of their service to avoid single point of failures and can be installed in the same type of node as the protocol servers.

To summarize the features of the front end zone, we have the following features:

- Front end servers are only there to process requests. They hold no user information and no specific techniques are required to share or duplicate data.
- The front end zone is basically comprised of a set of servers that run fully independent with virtually no coupling at all.
- Horizontal scalability is simply achieved by adding additional front end servers to provide more power to a given service.
- High-availability is a guaranteed because several servers are available to perform the same tasks.
- A security advantage that comes with the proposed architecture is that any intrusion on a given server will limit the harm to the service hosted on this server, and, assuming that several servers are used to provide the same service, the service will remain available (potentially in degraded mode).
- Horizontal scalability and high availability requires some advanced load balancing techniques. To gain effective load balancing and scalability, the load balancing technique must guarantee that:
  - All servers implementing a given service are used at the same time and that the load is evenly distributed on the set of available servers.

- To simplify configuration, the set of servers that implement a given service should appear as one server to the users.
- The server group for a service is fully dynamic:
  - A server failure is automatically detected. This server is automatically removed from the pool of available servers and new requests are assigned only to available servers.
  - If a new server is added (or a server comes back in an available state), then it should be dynamically discovered, and requests should flow to it automatically.

- In addition to horizontal scaling and high-availability, implementing services with a set of load balanced servers gives a lot of flexibility for scheduled maintenance activities. Stopping a given server is handled as a server failure and the platform will start to operate in degraded mode, but will still fulfill the service.
- Another advantage of designing the ISP platform in zones and isolating services on different servers is the possibility of controlling the performance of the platform on a very small granular scale and in a very controlled manner, for example, e-mail.

  There are several methods for an end user to read their mail. POP3 is the most classical one, WebMail is a very strong trend, and IMAP4, at least in the residential market, can still be considered as a differenciator, because few ISPs are offering it. Assume that as an ISP, you are only providing POP3 accesses and you decide to provide IMAP4 as a new option as part of a 'premium' offer. You want this service to provide extremely good performances, because you are pricing it higher than your basic POP3. Running specific IMAP4 servers, deciding how many you will need, and adding additional servers as the service develop is the right manner to introduce this service in a controlled and cost-effective way.

2.5 Back end zone

The next layers in our seven layer infrastructure are associated with the back end of the infrastructure.

We will take a look at the abstraction layer, which is the cornerstone of all interfaces to other components within the ISP platform. This layer in particular is not clearly defined as being front or back end, as it is dependant upon the middleware capability of the platform. The middleware of the platform applies throughout the platform. As such, services can be added both in the front end and back end zone. We will then take a brief look at the other layers, peeking

at the standards that can be employed to deploy effectively business and data access components. Finally, we will describe the benefits of back end zone configuration and its contributions to the nonfunctional requirements described in Section 1.3.2, “Architectural evaluation criteria” on page 20.

2.5.1 Abstraction layer

Generic and specific interfaces are provided to build up customized functions on top of the ISP platform. The specific interfaces are component specific interfaces. A generic interface, the so-called abstraction layer, can be built over the products or component-specific interfaces.

The abstraction layer can be used to attach new services on the ISP platform. Moreover, these services can be fully integrated into the platform’s existing services. Thus, such services provided by partners or third parties can be encapsulated. The components of the abstraction layer actively contribute in making these services collaborate together. They also contribute in enabling rapid service development. Figure 27 identifies the specific interfaces that an abstraction layer could encapsulate and deploy for service integration. These are only examples, as other specific interfaces could be found, depending on the business requirements and the technologies used.

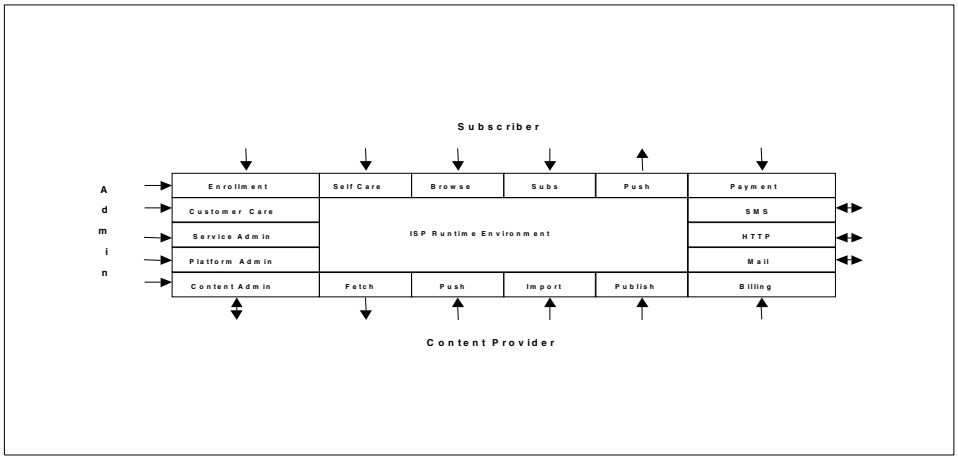


Figure 27. ISP platform interfaces

Subscriber interfaces

- Enrollment

  The Enrollment allows subscribers to register to the ISP.

- Self-care interface

The self-care interface allows subscriber to make their own service configuration and registering through different devices.

- Browse interface

  The browse interface allows users to browse the content and services an ISP provides.

- Subscribe interface

  The subscribe interface allows users to subscribe for content that is delivered from the ISP.

- Push interface

  The push interface sends messages to a subscriber from an ISP service.

- Payment interface

  The payment interface allows payments through Web browsers.

Content service interfaces

- Content import interface

  The content import interface allows third parties to publish content on the ISP platform.

- Content push interface

  The content push interface allows content to be pushed to the subscriber.

- Content publish interface

  The content publish interface allows third parties to publish content on the ISP platform and to forward it to the subscriber.

- Content fetch interface

  The content fetch interface allows the fetching of information from content providers.

Administration interfaces

- Customer care administrator

  The customer care administrator allows CSRs to manage subscriber data.

- Service administrator

  The service administrator allows administration of services and legacy applications.

- Platform administrator

  The platform administrator enables system administrators to manage the platform hardware, software, and network.

- Content administrator

  The content administrator allows management of the content that is available on the ISP platform.

Communication interfaces

- HTTP interface

  The HTTP interface allows the handling of HTTP requests and replies.

- SMS interface

  The SMS interface is the messaging interface.

- Mail interface

  The mail interface enables the sending and receiving of e-mail.

2.5.2 Application, persistence and data/content layers

The application layer holds the business logic that needs to be implemented in the diverse services of the platform. A number of standards that define a framework for application servers are described in this overview. The choice of technology we have made is Java-oriented, as this technology runs on all types of hardware. Other frameworks exist, but are not referenced here. Whatever the products and technology used, the procedures to complete the functions are similar.

The Web server is the component that handles service requests and replies. It forwards requests to the application server, which implements business logic for services. The application server is the integration node that has connections to different data sources, like database, service applications, and legacy applications. These connections define the persistence and data/content layer when connected to a database.

Web server

The Web Server handles the subscribers' HTTP requests and assembles pages, such as HyperText Markup Language (HTML), Wireless Markup Language (WML), or eXtensible Markup Language (XML), in reply. In order to achieve this in cases where business logic is required, the Web server accesses a variety of other systems through an integration node that is the application server. Typically, all static Web pages are directly accessed from the Web server.

Application server

The application server is an integration node that integrates back end systems. It provides the environment and infrastructure required to manage Java-based applications. The application server makes use of a Java runtime

environment or language independent Common Object Request Broker Application (CORBA) on the host machine.

Application servers host Java Server Pages (JSP), Java Beans and Enterprise Java Beans (EJB) that implement the ISPs service functionality and integrate the back end. The application server has interfaces to a number of services that are essential for the ISP.

CORBA

The Object Request Broker (ORB) supports interprocess object communication. The ORB supports a language neutral object model. The ORB enables:

- Object distribution by brokering method invocations to remote objects across processes or machine boundaries
- Object method invocations across different language environments
- Release binary compatibility so that parts of an application can be updated independently

Using the ORB, objects can be accessed independently of their location, across processes, or across distributed systems. The ORB utilizes many other services, including security, directory, and communication services, to accomplish its function.

Java RMI

Java Remote Method Invocation (RMI) is a set of APIs designed to support remote method invocations on objects across Java virtual machines. RMI directly integrates a distributed object model into the Java language such that it allows distributed applications to build in Java.

Java components that use the RMI interface can make call on a remote object once it obtains a reference to the remote object, either by looking up the remote object in the bootstrap naming service provided by RMI or by receiving the reference as an argument or a return value. Java RMI uses a combination of Java Object Serialization (JOS) and the Java Remote Method Protocol (JRMP) to convert normal-looking method calls into remote method calls.

Java Beans

Java Beans are Java component for representing and storing data. The beans can be of any type; from visual widgets to abstract business objects. Beans can be composed using a variety of techniques, including direct event connections, event handling through scripting, publishing of an inner bean's

application programming interfaces (APIs) through delegation, and publishing new APIs.

EJB

Enterprise JavaBeans (EJBs) are the new Java component model for enterprise applications. More than components, EJBs provide a framework. They combine Java server-side components with distributed object technologies: CORBA + RMI. EJBs are always distributed, which make them fundamentally different from standard Java Beans. EJBs provide the business logic part of an application. They are not concerned with presentation issues, and therefore must be used in conjunction with display technologies like Servlets or JSPs for HTML clients, or Java applications that use technologies like AWT or Swing. Application servers that implement the EJB specification can simplify business application systems by providing services that handle complex security, resource pooling, persistence, concurrency, and transactional integrity.

Internet Inter-ORB Protocol (IIOP)

With CORBA 2.0, the Object Management Group (OMG), which is the organization responsible for the CORBA standard, has introduced a new communication protocol called IIOP. In order for an ORB to be CORBA 2.0 compliant, it has to use IIOP as the communication protocol for inter-ORB interaction. IIOP is a TCP/IP-specific implementation of the GIOP.

The GIOP specifies message formats and Common Data Representation (CDR). As a result, the various ORB implementations can communicate across heterogeneous environments. The CDR takes care of any cross-platform data differences, such as “endianness” and floating point representation.

Java technologies have also adopted the IIOP as the communication protocol for Java components. It is possible to use Java RMI over IIOP and EJB uses IIOP.

JDBC

Java DataBase Connectivity (JDBC) is a Java API that enables Java programs to execute Structured Query Language (SQL) statements. API allows Java programs to interact with a SQL-compliant database. Relational Database Management Systems (RDBMSs) support SQL. Java itself runs on most platforms, so JDBC makes it possible to write a single database application that can run on different platforms and interact with different DBMSs.

For additional protocol information and descriptions, refer to Appendix A, “Network protocols” on page 283.

2.5.3 Physical implementation features

The back end zone contains all the applications working on user information. These applications need access to user data stored on disk and database. Only one instance of the database should run in the back end zone. One application type runs only once in the back end zone. It can not be load balanced as in front end protocol servers. Instead, to guarantee high availability and scalability, a clustering of the application must be achieved in a takeover mode. If one machine or application fails, it should be automatically detected. The same application should then be launched on another machine in a stand by mode.

This configuration is less flexible than for front end servers. This type of node must be more powerful, with more CPU and memory, to be able to access the disk and manage the context of each user connection. The machine power should also be chosen to withstand traffic growth, without any intervention/upgrades, to guarantee high availability of the service. Such upgrades of machine, when necessary, should be done outside peak hours and in a predictable time frame.

A centralized disk is installed. It centralizes the administration of the disk and enables take over recovery. Such considerations are addressed in Section 2.6, “Storage zone” on page 82.

Finally, from a security perspective, back end servers that house subscribers profiles should be located in a different subnetwork that can be accessed only from the front-end servers.

2.6 Storage zone

The storage market has experienced a significant growth during the past few years. The technology has made some giant steps forward enabling more storage capabilities at a lower price.

In the service provider segment and in particular for the emerging ASP segment, storage is the cornerstone of an infrastructure. Storage Area Network (SAN) promises to solve the difficult issues ISPs face due to their demanding financial and business requirements.

We will look at the evolution of storage and define what a SAN is. We will then look at the ISP requirements SAN meets.

2.6.1 Storage evolution

From mainframes, PCs, and UNIX stations to SAN, we look at the storage technologies IBM has used during the past decade.

Mainframes

Storage devices were first directly attached to servers. Servers were loaded to control storage activity and a storage device was dedicated to a single server. Subsequent evolution led to the introduction of control units. Control units are storage off-load servers that have a limited level of intelligence and are able to perform functions such as I/O request caching for performance improvements or dual copy of data (RAID 1) for availability.

In 1991, with OS/390, IBM introduced new form of connectivity between S/390 servers and storage devices called Enterprise System CONnection (ESCON). Based on optical fiber, ESCON architecture introduced in time dynamic connectivity, a higher speed of transfer, and longer distances between server and storage devices. With global management capability, ESCON represents the first SAN architecture.

The S/390 FICON architecture is an enhancement of the existing ESCON architecture. Among other benefits, FICON architecture enhances data rates, distances allowed, and number of devices connected.

PC and UNIX

Different approaches to connectivity were taken for PCs and UNIX stations. For PCs, the AT Attachment (ATA) interface, comprised of Integrated Drive Electronics (IDE), Enhanced IDE (EIDE), and Ultra ATA (UltraATA), was first introduced. Then, the parallel interface Small Computer System Interface (SCSI) was conceived to allow up to 16 devices up to 25 meters apart to be connected.

Although SCSI protocols can be used on Fibre Channel (then called FCP) and SSA devices, most people mean the parallel interface when they say SCSI. The SCSI devices are connected to form a terminated bus (the bus is terminated using a terminator). The SCSI protocol has many configuration options for error handling and supports both disconnect and reconnect to devices and multiple initiator requests. Usually a host computer is an initiator. Multiple initiator support allows multiple hosts to attach to the same devices and is used in support of clustered configurations.

Serial Storage Architecture (SSA)

Serial Storage technology was introduced by IBM in 1991. Today, IBM's third-generation Serial Storage Architecture (SSA) solutions provide an open storage interface designed specifically to meet the high performance

demands of network computing. SSA is primarily an architecture for device attachment within subsystems, but it is also used connect devices to servers. When compared to other technologies, such as parallel SCSI, SSA offers superior performance and data availability with greater flexibility and value.

SSA uses a loop rather than a bus. The SSA architecture enables two simultaneous read paths and two simultaneous write paths; each of the four paths has 40 MB/sec bandwidth. With arbitrated subsystems, such as SCSI and Fiber Channel-Arbitrated Loop (FC-AL), arbitration occurs before and after every transfer. This significantly reduces the amount of data that can be transferred. Even 100 MB/sec FC-AL can sustain significantly less than 50 percent of the available bandwidth for transaction processing systems. SSA has no arbitration and can utilize the available bandwidth significantly better than SCSI.

With the SSA loop design, a single cable failure will not cause loss of access to data. If there is a failure in the loop, the SSA adapter will automatically continue accessing the devices in a non-loop configuration. Once the path is restored, the adapter will automatically reconfigure to resume the normal mode of operation. If there is a disk failure, the hot-swappable disks can be removed without loss of communication between the adapter and the other disks on the loop. Add disk drives and, depending on access patterns and rates, performance increases; add adapters, and performance increases; segment into additional loops, and performance increases again. These types of performance increases do not occur in SCSI architectures.

2.6.2 Storage Area Network

We will now look at the needs that SAN addresses and its features.

Requirements

SAN architecture is meant to address needs such as:

- Heterogeneous access to data

  Support for many distinct operating systems and servers is a must for service providers. This is especially true for ASPs, which necessarily provide a range of applications to their customers that will most likely run on distinct operating systems.

- Automatic backup, archiving and recovery

  These are ideal requirements for service providers who require 24x7 availability. Implementing automatic operations guarantees the business continuity of the service delivered, even in case of failures.

- Business continuity during maintenance

Hot pluggable devices enable maintenance, addition or removal of disks while the platform is operating. Failure of a disk does not impact other devices on the network.

- Seamless evolution on demand

  The infrastructure of the storage zone must enable evolution without interruption of service. This means that the infrastructure itself must be able to evolve.

- High speed data rate

  With the explosion of data on the Internet and the centralization of information in huge databases, speed rates become a crucial factor in the viability of an infrastructure.

- Direct access from multiple servers

  Sharing a disk between multiple servers guarantees redundancy in the service delivered, and is therefore an important component of the high availability of the infrastructure.

ISP requirements are even more drastic when it comes to evolution. Reactivity is a crucial capability in that segment, as traffic volume and subscriber base growth are often unpredictable. Successful ISPs will face issues in adding more services very quickly and coping with subscriber base growth very quickly. SAN architecture addresses such requirements.

Features

SAN features are indeed:

- Smart storage and smart channels

  New devices gain intelligence and storage capability. With these new decentralized intelligent storage devices and new connectivity means, SAN can expand to be comprised of these devices. This is the possible future for SAN architecture.

- Fat pipes and long distances

  The volume of data grows exponentially world-wide. SAN is based on Fibre Channel technology. Fibre Channel is not only optical fiber; it is a generic name that designates the connectivity between servers and disks. New technology, such as laser transmission, is promising longer distances between servers and disks and much higher data rates.

- Any to any access

  This is when all operating systems can access the same data whatever the format. This is not a reality at the time of this publication, but it is a goal for the future.

- A single point of control

  SAN architecture allows a central management of distributed systems, where backup is addressed as being the major issue. This is especially important for geographically distributed systems. The management console has to be unique for all attached devices.

- True incremental growth

  When mixing technologies, the limits are boundless when it comes to growth. SAN architecture enables you to start small and then evolve to bulk storage solutions incrementally and seamlessly.

- Investment protection

  As technologies evolve, new devices and protocols can be incrementally added to the SAN architecture without having to replace existing devices.

- Industry standards

  This is still an open area that is the focus of the efforts of many organizations and industries.

SAN architecture

SAN technology is still evolving, so it is relatively expensive. We recommend deployment of small SAN as support backup *only* at this time. When the cost of large SAN switches is more economic, a wider use of SAN should be considered.

A storage management server within each data services building block is attached to the SAN. This server is able to access all disk storage within its data services building block. A shared tape library within common and network services is also attached to the SAN. This enables data within any data services building block to be backed-up to a tape library without impacting LAN performance. A library manager server is deployed within common and network services to coordinate shared access to tape library. This is represented in Figure 28 on page 87.

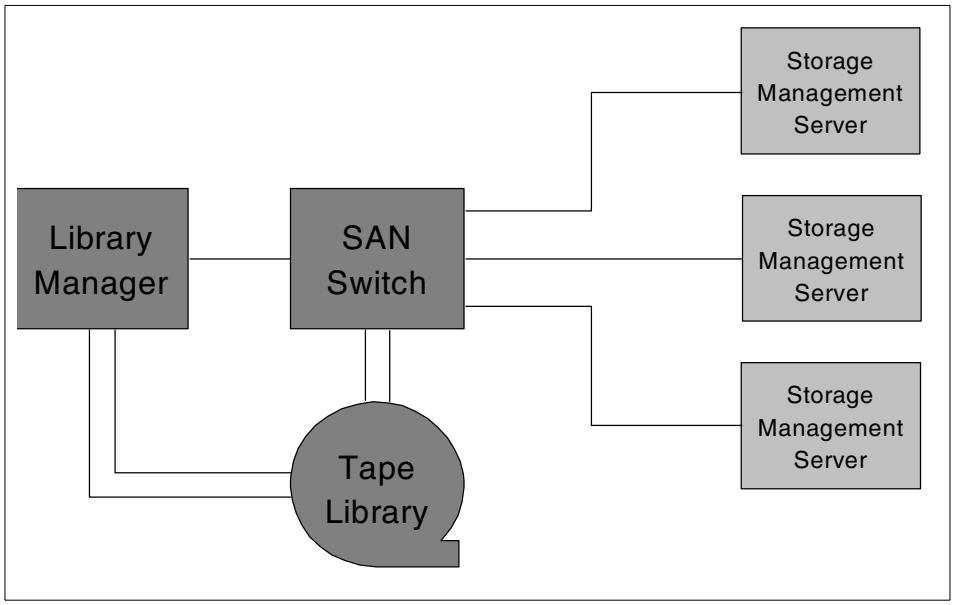


Figure 28. SAN design concept

2.7 Management zone

The management zone provides the set of tools available to administrate the ISP platform. An ISP infrastructure can be geographically dispersed among a a great number of machines or simply in a highly complex nature. This is especially true when there is more than the ISP platform or when the company owning the platform has its own IT infrastructure. Efficiently managing this infrastructure is a key element in maintaining its integrity and appropriate service level. Implementing administrative tools to monitor such an environment is out of scope of this book. Products that could be used in that environment are described on Section 3.7, “Platform administration” on page 164.

The administrative requirements for a single ISP platform are described in this section. The layers of administration are:

- Subscriber management
- Service management
- Software management
- Hardware management

- Network management

Subscriber and service management are strongly related to the ISP environment. Users enroll to the ISP and become subscribers. In doing so, they choose a deal. In choosing this deal, they choose services related to this deal. Subscribers have access to potentially different services defined by deals, which are defined by the subscriber management administrator and stored in the subscriber base. Billing and customer care are crucial components in that space. We describe both functions in Section 2.8, “Legacy zone” on page 94. The most common legacy applications to integrate with the ISP platform are existing sales channels, customer care tools, existing billing engine, and the customer database.

We look at each basic service delivered by the subscriber management and how they can be configured and customized. We look at the business models that can be implemented and how they can affect the organization of the subscriber base. We will also look at the security impact that has to be handled by the administrator. Software, hardware and network management are also described (mostly in regards to the RS/6000 SP, due to its unique management features) and in regards to the network components we have previously described.

2.7.1 Subscriber management

The subscriber management is the management of the customer care environment. We define the parameters of a customer care solution for an ISP highlighting the management capabilities expected.

Complex subscriber management can cluster and handle many domains within the same base. This implies that there is an embedded management for each domain and that a super-administrator has the ability to globally and centrally manage the given domains.

Virtual ISPs
A subscriber base that is fragmented into many domains is the business model of a Virtual ISP and/or Virtual Portals. This model is on the boundary of the *Packagers* and *Hosting Services* ISP model, which is described in Section 1.2.4, “Segmentation” on page 4. This model is a specific type of hosting ISP, which is a shared service model. The shared service model allows multiple customers' Web sites to reside on a single server. The servers are owned by the hosting ISP and the customer pays for the hosting service.

This model can be implemented in the same kind of platform as an access ISP architecture, but the operations around the ISP platform may vary.

Indeed, customer care may be handled either by the ISP or by the customer. In both cases, the infrastructure and the definition of the operations differ from the traditional access ISP.

Domain management
Domains must be managed by the super-administrator, who should be able to edit, add, and remove domains. Additionally, a super-administrator should be able to define the access controls to each domain, such as configuring security profiles for CSR (see Section 2.3.3.1, “Enrollment” on page 66). This configuration ensures that each domain is sufficiently clustered to safely separate business data within a shared server.

Each domain is a logically separated subscriber base. All settings within this base should be manageable independent from the others. A domain must edit, add, and remove domain components, such as:

- Billable residential subscriber accounts
- Billable business account for many subscribers
- Deals to choose from

ISP services
We have considered the services described in Section 2.3.3, “ISP services” on page 66. These services are the built-in components of the subscriber management component. Other services that may be integrated are considered in Section 2.7.2, “Service management” on page 89.

The registration configuration of an user self-enrollment involves defining an enrollment interface. This enrollment interface may vary, depending on the domain. Each domain may require its own look and feel, but more importantly, different settings. The subscriber management tool should enable the administrator to customize the settings for the enrollment interface for each possible domain. A global enrollment interface should also be applicable. The customer self-care interface can use the same settings, which are handled in the same manner.

The personalization configuration is strongly related to the look and feel of the portal pages of the domain. These can be handled by a Webmaster looking after the JSPs, but may also require an administrator with Java skills to maintain the business components responsible for personalization.

2.7.2 Service management

Service management is part of subscriber management. We will consider services that are not built into the subscriber management suite. Such

services, which range from a mail system, a video streaming system, and an audio streaming system to an online shopping mall or a content base, are external software components that need to be integrated within the subscriber management environment.

Billable components
These services are potentially billable services.

These services can be proposed to a subscriber as, for example:

- Inside different bundles offered at a fixed price
- Billed separately on usage
- Billed on volume used

These services are components of a deal. When a user subscribes to an ISP, he chooses a deal. This deal determines what service the subscriber will be able to use and under what conditions.

Provisioning
The subscriber management needs to record which services a subscriber is authorized to use. Some of these services may need to be automatically provisioned. Examples of this include:

- If the deal enables the subscriber to have mail on the platform, a mailbox will be automatically created.
- If the deal enables the subscriber to have personal pages on the platform, the subscriber management will need to automatically reserve some disk space.

Authorization
The deal contains the services a subscriber is allowed to use. The subscriber management tools should be comprised of an authorization checker that checks whether a subscriber can access a service upon request. This authorization checker can be a third party component that would need to have access in some way to the deals and user profile in the subscriber base or in the LDAP directory. The subscriber management should be able to access, in real time, an LDAP directory for integration purposes.

Single sign-on
Single sign-on is a property that enables a subscriber to access each service he is entitled to without reentering a user name and password. In other words, once subscribers have been authenticated on the ISP platform, they do not need to authenticate themselves with the services at hand. The state-of-the-art single sign-on is to enable it for services that are not on the

ISP platform but distributed in other locations, like service or content partners on the Internet.

This feature is important for an ISP and requires integration work between software components. Again, the subscriber management needs to handle services that can be geographically distributed with central deal components.

2.7.3 Software management

Software management relies fundamentally on three aspects:

- The specific software installation and management features.
- The hardware it is installed on.
- The architecture scalability and high availability features.

By software management, we mean the ability to install, configure, upgrade, enable, disable, and audit software components. These abilities are mostly defined by the features offered by the product. The ideal way to manage a suite of software is to be able to manage it centrally from one single workstation. This becomes a challenge when the hardware products used on the platform are heterogeneous. Having a geographically distributed environment makes matters even worse. An example of a product suite handling such requirements is given in Section 3.7, “Platform administration” on page 164.

The hardware components can be crucial in making it easier to manage the software. The RS/6000 SP offers features that significantly enhance the software management. The crucial feature delivered is the ability to clone the installation of software components from the Control Work Station (CWS). An overview of the RS/6000 SP manageability can be found in Section 2.7.4, “Hardware management” on page 91 and in Section 4.1.4, “Manageability” on page 172. The software management capabilities of the RS/6000 SP is demonstrated in Chapter 6, “Sample implementation” on page 219.

Finally, architecture scalability and high availability is an important factor in the management of a platform and its software components. The ability to take any node in the system off-line makes it easier to do any type of operation on the software.

2.7.4 Hardware management

Hardware management is specific to the management capabilities of a specific suite of hardware components; we will consider the RS/6000 SP and its AIX Operating System (OS) here.

The ISP platform is ideally managed within the RS/6000 SP, as described in Section 4.5, “Benefits of using the SP in the ISP arena” on page 193. We give here an overview of the RS/6000 SP’s management capabilities.

IBM RS/6000 systems are the proposed hardware that use IBM AIX V4 as the operating system. In the ISP case, the RS/6000 Scalable POWERparallel Systems (RS/6000 SP) is proposed. The RS/6000 SP can be composed of up to 512 nodes, each of them being a complete RS/6000 system with AIX. The RS/6000 SP system can be managed from one central Control Workstation (CWS).

For installation over a network, AIX has an integrated tool, the Network Install Manager (NIM). An automatic installation service can be defined with NIM. If the system administrator has properly defined NIM, all new systems can be connected to the net and will automatically install AIX after power-on. The SP system also uses NIM to install the nodes from the Control Workstation.

RS/6000 and AIX are designed for High Availability (HA). All integrated extensions are designed to minimize and avoid downtime. The Dynamic Resource Manager (DRM) in AIX can activate new devices without the need of rebooting the computer. The Journaled File System (JFS) ensures file consistency, so no long-lasting checks on file integrity are needed.

The System Management Interface Tool (SMIT) is a menu-driven system tool for AIX. All system management tasks can be performed by means of SMIT. SMIT uses AIX high-level commands offered to the user as menus. All activities performed via SMIT are logged into script- and logfiles, which can be used to create shell scripts easily. SMIT offers a graphical interface and a Web-based subsystem, so that the system can be managed by a browser over the intranet.

The whole storage management in AIX can be performed with the Logical Volume Manager (LVM), a set of operating system commands and tools that can also be used by SMIT. Storage management includes activating disks or disk subsystems, such as a Redundant Array of Independent Disks (RAID), and creating volume groups, logical volumes, and file systems.

AIX uses the concept of Logical Volume Storage to manage the disk storage. This method divides and allocates storage space on a system disk. It allows system administrators to construct a more logical view of physical storage. The logical view of physical storage allows the configuration of the storage system for higher availability and better performance, as opposed to just plain physical storage. It provides flexibility in storage administration, such as the

configuration of the number of copies, location, and the size of logical storage units.

To increase the availability of data, file systems can be mirrored over more than one physical disk. AIX includes support for mirroring. Disks can also be connected to two or more RS/6000 systems; therefore, sharing of disk storage is achieved. Access and ownership can be controlled by AIX mechanisms. This feature is used by HACMP, the High Available Cluster Management Software from IBM, which is described in Section 4.4.5, “High Availability Infrastructure (HAI)” on page 185.

AIX has tools to create a system backup. A tape backup from the system is bootable and can be used to recreate the whole system in a short period of time. It can also be used to clone systems. This means that you can install a system with all software and device-drivers that are needed, and use the backup to install other computers with equal hardware. This backup can also be used by NIM to install systems over the LVS.

2.7.5 Network management

The network management is the management of the network components of the ISP access network environment. The main network components are:

- Routers
- NAS farm
- RADIUS server
- DNS

Routers can be administrated remotely with specific administration tools. Router configuration is crucial for the ISP platform’s well-being.

NASs are organized in farms. NAS farms can be managed separately from the ISP platform. Management of a NAS farm involves checking that these NAS features are up and running:

- Handle Point to Point Protocol (PPP) and Serial Line Internet Protocol (SLIP) with the terminals
- Handle IP protocol with the platform router
- Responsible for handling the following information and appropriate routing:
  - Phone circuit number
  - IP address that is allocated to the terminal
  - Service provider identifier

- Split control (identification/authentication, accounting) and data traffic
- Support RADIUS client function

The RADIUS server is an important component of the ISP platform. Not only is it the vector through which the user is authenticated, but it accounts for the time spent by subscribers on the platform. It is, in that sense, a billing tool. The settings of the RADIUS server should be configurable for the subscriber management tool.

The DNS servers are redundant on and outside the platform. These components detain sensitive information (names and IP address of the platform machines) and are deployed accordingly.

2.8 Legacy zone

Legacy applications are existing applications that are possibly located on the ISP's intranet or specific solutions for special needs. Billing and Customer Relationship Management (CRM) solutions are examples of legacy applications. These could be applications used in a previous or existing business done by the ISP. These applications, depending on the business model of the ISP, need to be integrated with the ISP platform to allow the ISP to go in production. Not all ISPs have legacy applications (start from scratch) or need to integrate their legacy applications with the ISP (separate accounting and sales channel, for example).

Indeed, when traditional companies choose to offer an ISP service either to their existing customers or as a new line of business, they usually have an existing IT infrastructure. Companies in the Telco or Media segment are enterprises which are likely to have:

- One or many existing sales channels

  Sales channels may vary from customer care hotlines to existing Web sites for registration.

- One or many billing systems

  Configurations may vary from a centralized billing system for all services delivered to distinct billing systems for each service delivered.

International companies are likely to have distributed sales channels and billing systems.

The legacy layer represents those applications that can be integrated with the ISP platform. The decision to integrate such applications depends heavily on the business model of the company.

We look here at distinct generic business models in which such integration may be necessary. These models are only for illustration purposes. Some patterns are extracted. We detail what needs to be achieved in order to comply to the given case requirements. We also detail what are the requirements regarding billing methods and how these requirements can be implemented.

2.8.1 Business cases

We look here at business case samples, which are representative of the issues and operations an ISP is likely to face in order to go in production with the legacy applications.

Business model

The usual model that involves sales channels, account activation, and billing is represented in Figure 29 on page 96. This model accounts for an integrated ISP platform, where:

- The user can subscribe from the Internet or from customer care channels.
- The subscriber account is automatically created, fully provisioned and activated.
- The billing account is created for the new subscriber and credentials are checked for.
- Usage is collected on the billing engine for processing.

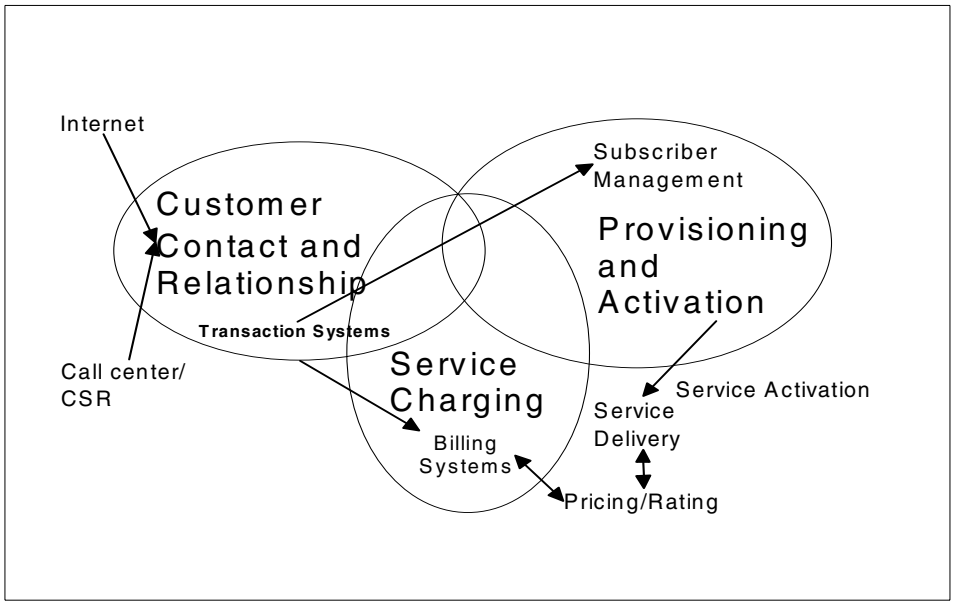


Figure 29. Sales channels and provisioning model

This is the model an ISP has to achieve in order to go in production. Existing systems that manage Customer Relationship Management (CRM) and billing operations need to be integrated with the ISP platform to comply with the features of this model.

Operations

There are two levels of integration that must be addressed:

- Setup of the ISP platform
- Every day processes definition and implementation

The setup of the platform refers to provisioning the system with the initial settings and parameters. As part of this setup, the ISP may want to deliver Internet and ISP platform services access to an existing set of customers. (We call this operation bulk enrollment.) Businesses like mobile companies, traditional service companies, Telco companies, or Media companies will want to offer their customers access to a new ISP service, whether or not it serves their existing line of business.

An every day operations definition give a concrete look at:

- What are the sales channels for subscribing to the ISP platform?
- What customer care tools are to be used in order to subscribe to the ISP?

- Which billing system will be responsible for issuing the subscriber a bill for its platform usage?

The business requirements and the effort and cost linked to the integration work are the major factors that help define those operations.

Bulk enrollment

Bulk enrollment relies mostly on the capability of the subscriber management tools of the ISP platform to allow creation of subscribers from an external source. Bulk enrollment is a one time operation. The process used to transfer data from an external database to the ISP platform subscriber base can be automated and reused in other cases.

Leverage existing sales channels

If existing sales channels are to be used to sale Internet and ISP platform services access, there are two options that can be used to transfer new subscribers to the ISP platform subscriber base:

- Real-time transfer
- Batch transfer

A real-time transfer can be implemented from the existing customer care engine or from the database the request is stored on (using a trigger of some sort). In both cases, the transfer must be handled in a transaction. Real-time transfer allows the user or the CSR to act immediately upon problems.

Batch transfer can be implemented from the database the request is stored on. This is a daily automated operation which will fetch new requests and send them to the ISP platform subscriber management for registering. The operation can be done in a similar way to the bulk enrollment described above. The transfer should also be done in a transaction.

In the case where more than one database is used to store such requests, there is a chance that subscriber data could overlap, making it impossible to create a subscriber account (for example, two subscribers with the same name from different regions or countries). This is why a transaction is necessary to transfer the data. The granularity of the operation should also be considered in that respect.

Distribute ISP customer care tools

The existing sales channels can also be leveraged to use the tools provided by the ISP subscriber management suite. Many factors should be considered in order to achieve this:

- CSR location and connectivity possibilities

- Existing Internet pages leveraging

CSRs can be geographically distributed. If they are to use the ISP platform customer care tool, connection to the ISP platform subscriber management must be enabled. This can be complex from a security perspective, depending on the company IT infrastructure. CSRs must be educated with the new tool.

Existing Internet pages that would be used in the line of business of the company must be leveraged to enable the subscriber to register.

Integration
Integrating the ISP platform within the operations of the enterprise is clearly complex. The objective is to achieve the model represented in Figure 29 on page 96. We have described some examples in this section that are not complete, but which give a flavor of the necessary tasks that have to be accomplished versus the level of integration desired and the IT infrastructure in place.

The subscriber management must be open to allow different means of enrolling subscribers than the ones provided in its basic service. The middleware on which the subscriber management is based should extend its capabilities rapidly and offer strong transaction support.

2.8.2 Billing systems

Whether a new billing system is provided within the ISP platform or whether an existing billing system is used from the IT infrastructure, the processes and functions are similar. Only operations may differ depending on the complexity of the overall component distribution and number. The description found under this section apply for both cases. For the sake of simplifying the overall picture, we describe the new billing system case.

ISP revenue sources
Here we look at the traditional and new revenue sources for an ISP and at the way it impacts the ISP architecture, depending on the type of billing they choose to implement.

ISPs have used several sources of revenue in their business models, and the models are still undergoing evolution. Nevertheless, we can outline the various basis of revenues as:

- Flat fee/flat rate

  A subscriber pays a fixed amount of money for the Internet access service. In some countries, local communications from the Telco network access

point (home/office) to the ISP are chargeable on top, or some duration is included in the flat fee from the ISP. Other flat rate schemes include PC or other equipment rental with the Internet connection.

- Usage based

  A subscriber pays for connection time or data volume (IP packets).

- Advertising based

  The ISP charges advertisers for a spot on their access panels.

These sources of revenues are evolving, as access tends to be provided for free, and more services are made available by ISPs that "walk-up" the value chain, for example:

- Games or on-line services.
- ISPs providing search/portal services; pay for the reference or get a portion of a transaction fee.
- Hosting Web pages/sites.

The critical success factors are the:

- Number of subscribers (obvious for flat rate revenues)
- Hits on site (to attract advertisers)
- Time spent on site (to attract advertisers or to generate time based revenues)
- Services provided (value chain revenues)
- Competition
- Subscriber churn rate

Billing methods

The billing offerings have evolved since the first ISPs have been implemented. The issue is not being on the market and looking good, but to make profits. Old and new ways of billing customers for ISP services reflect the new trend. A number of methods are:

- Dial up connection time

  This is the traditional way of billing Internet access. The main operations required are:

  - Rating

    Deals need to be assigned billable information: cost for x hours of connection with initial free hours or days and cost for hours over subscription.

- Usage gathering

  The billing engine computes usage hours of connection for each subscriber.

- Billing management

  At the end of the billing period, the billing engine provides a bill according to the subscription price and the transaction time spent by the user. The user can display detailed information concerning the bill: transactions included in the billing period, with time spent in all transactions and payments done.

The RADIUS accounting is sufficient for recording such information as connection and disconnection of a subscriber. The RADIUS server can record those on the subscriber for auditing and billing post treatment. RADIUS is described in Section 2.1.2.1, “NAS and authentication servers” on page 39.

- Pre-paid

  In addition to the operations described above, new operations need to be implemented:

  - Credit management

    This solution allows the possibility for the user to know the remaining allowed time and to buy additional hours of connection with new payments.

  - Pre-payment management

    When the allowed connection time is reached, user connection is interrupted. Users can be notified of the interruption prior to disconnection.

  The RADIUS accounting is not enough to support pre-paid services, as real-time billing and events are required, such as disconnecting a subscriber when he reaches the end of his credit line. An IP mediation device may be required to achieve such functions. It would then be integrated with the subscriber management or billing engine.

- Service based usage and volume

  A new way of gaining a high profit margin is to charge for additional services over the basic traditional ISP services.

  - Service management

    Users can select deals where they gain access to additional services. These services can also be provided outside the deal settlement and charged on top of the flat access rate. These ISP services are charged

on a time or volume usage. When the user selects the service, measurements are done on the connection time or the volume of data downloaded. Such management requires an IP mediation device.

Billing architecture

For complex measurement of user activity, a fully integrated solution is necessary. Basic features are required to support the ISP minimum requirements. We present in Figure 30 a logical architecture of an integrated billing solution that covers the main features of all three methods of billing described above.

A mediation device is a device that is generally able to gather and summarize usage records by user, account, or service type. A pure billing engine will use the IP mediation device as a feed.

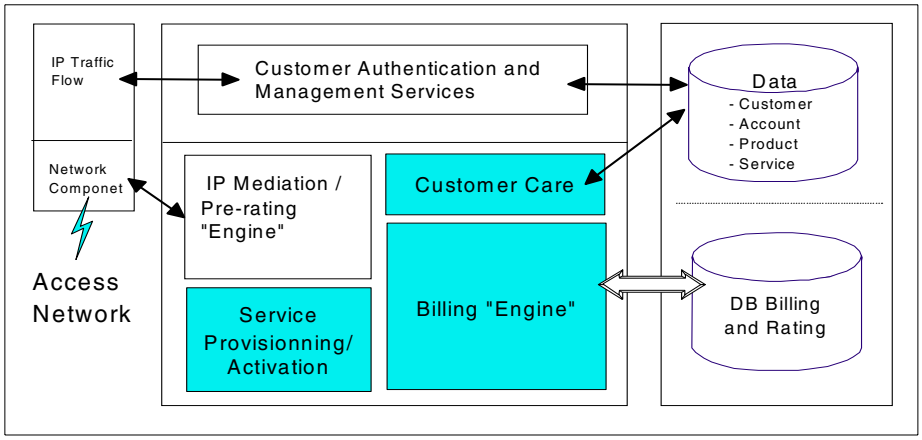


Figure 30. Billing logical architecture

The IP mediation/pre-rating engine is closely coupled with one or many network components. Such network components can be:

- A router

  The ISP access network router has the ability to unstack the TCP/IP frame, thus identifying the originator of the packet and the destination address of the server. It accounts for all IP traffic on the platform. Once integrated with an IP mediation device, it can deliver all information required to the IP mediation device to support service based usage and volume. The IP mediation system can sort and collect the raw information and feed it to the billing engine, or store it in an account database for the billing engine to collect later (depending on the requirements versus real time processing).

- A NAS

  The IP mediation system is required to, in real-time, cooperate with a billing engine or pre-rating engine to measure the authorized time left for a user who has a pre-paid account and take actions on specific events (five minutes left on your account, recharge your account, or disconnect). The ability to shutdown the connection in real time can be applied on the NAS. Depending on the capabilities of the IP mediation device and the billing system, some integration work may be required.

The IP mediation/pre-rating engine must also communicate with the subscriber management to enable deactivation of the account in real time. The next time users log in, their account is deactivated, if they have reached the end of their credit.This may require integration work depending on requirements.

Services which are charged for may also be required to communicate with the IP mediation device. The ISP access network router may deliver the IP address of a server, but a server could host more than one service. Integration work, depending on the service deployment, business requirements and capability of the application, may be required.

ASP market

Today, with the newly emerging ASP market, applications are likely to be charged for usage. ISPs must develop applications that are flexible enough to be integrated with a billing system or mediation devices. A framework that provides strong transaction and billing API support is strongly recommended to implement such applications.

Chapter 3. Components of an ISP environment

This chapter contains descriptions of a number of IBM and IBM Partners software products that can be used to construct an ISP platform and its numerous potential services.

First, we look at the components that can form the root of an ISP platform. We are using WebSphere EveryPlace Suite (WES), which is an offering based on diverse software components integrated with one another to provide connectivity, content adaptation, security, optimization, management capabilities, and basic services. We also take a look at a set of firewalls that can be used efficiently with WES.

We then go through an ordered list of software and solution offerings from IBM and IBM Partners. This list is not thorough in regards to what IBM or IBM Partners have to offer. The choices of products described here were motivated by the pertinence of their feature in covering ISP requirements and the available material the authors had at the time of this redbook. The level of explanation may vary between products as once again, the level of documentation at hand varied.

IBM Partners are very important for IBM in the ISP/ASP arena and, more specifically, in the NetGen business. IBM has positioned itself not as an Independent Software Vendor (ISV) but as a solution-based integrator.

IBM has created the ISV Center for NetGen. This is one of a series of initiatives to help NetGen companies in such areas as ISPs, ASPs, commerce sites, content Web sites, B2B portals, and e-MarketPlaces. Building an e-business Web site usually requires the integration of a large number of products from distinct vendors. The ISV Center for NetGen is aimed at enabling this integration. The center is used as a permanent showcase infrastructure to help NetGen customers build and validate personalized solutions based on market leading applications on IBM platforms. It is a permanent showcase that brings industry-leading IBM and ISV partner technical expertise together to accelerate the selection and integration of an end-to-end IT infrastructure. The ISV Center for NetGen also provides NetGen companies with a single unique point of access to a wide range of best-of-breed applications that they need to best match their unique IT infrastructure requirements.

Finally, the center provides NetGen companies with access to a wide range of IBM technical skills, and leverages more than 250 specialists from the advanced technical support centers.

The Partner products described here are a subset of a number that the center handles.

3.1 Firewalls

Firewalls are used for protecting the ISP's internal assets and customer data. The basic principle is that any publicly accessible system is assumed breakable, and is therefore placed in a separate zone from the internal, trusted network. We propose two firewalls that can be used to protect the ISP platform:

- One between the Internet and a DeMilitarized Zone (DMZ) that handles incoming requests from the Internet.
- One between the DMZ and the internal network containing the application nodes.

Routers can have built-in firewall capability and should be considered as a reliable and efficient way to secure the ISP platform.

3.1.1 IBM SecureWay - First Secure Boundary Server

SecureWay Firewall V5.1 enables a safe and secure platform by controlling all communications to and from the Internet. The IBM firewall contains all three critical firewall architectures: they are filtering, proxy and circuit level gateway.

Network Security Auditor proactively scans the firewall and other hosts to find potential security exposures. The SecureWay Firewall includes VPN support based on the IPSec standard. It also provides seamless Internet access using standard client software, and it supports TCP and UDP applications through SOCKS Version 5.

An administrator can centrally manage and configure multiple firewalls. The SecureWay Firewall provides also real-time performance statistics and log monitoring.

The SecureWay Firewall runs on AIX and NT.

More information on this subject can be found at:

`http://www-4.ibm.com/software/security/firewall/`

3.1.2 CheckPoint Secure Firewall-1

FireWall-1's scalable, modular architecture enables an organization to define and implement a single, centrally managed Security Policy. The enterprise Security Policy is defined at a central management console and downloaded to multiple enforcement points throughout the network.

CheckPoint Secure Firewall-1 architecture overview

FireWall-1 consists of the following components:

- Graphical User Interface (GUI)
- Management server
- Firewall module

The firewall module is deployed on Internet gateways and other network access points. The management server downloads the security policy to the firewall module, which protects the network. The firewall module includes the inspection module and the CheckPoint FireWall-1 security servers.

CheckPoint Secure Firewall-1 features overview

The security servers provide authentication and content security features.

- Authentication

  The security servers provide authentication for users of FTP, HTTP, Telnet, and Rlogin. If the security policy specifies authentication for any of these services, the inspection module diverts the connection to the appropriate security server. The security server performs the required authentication. If the authentication is successful, the connection proceeds to the target server.

- Content security

  Content security is available for HTTP, FTP, and SMTP.

- HTTP

  The HTTP security server provides content security based on schemes such as HTTP, FTP, or Gopher, methods, such as Get or Post, and hosts, such as "*.com", paths, and queries. A file containing a list of IP addresses and paths to which access will be denied or allowed can be used.

- FTP

  The FTP security server provides content security based on FTP commands, such as Put or Get, file name restrictions, and anti-virus checking for files transferred.

- SMTP

  The SMTP security server provides content security based on From and To fields in the mail envelope and header and attachment types. In addition, it provides a secure sendmail application that prevents direct online connection attacks. The SMTP security server also serves as an SMTP address translator, that is, it can hide real user names from the outside world by rewriting the From field, while maintaining connectivity by restoring the correct addresses in the response.

More information on this subject can be found at:

`http://www.checkpoint.com/`

3.2 ISP basics with WES

An ISP platform provides an infrastructure for network connection and ISP services, and, these days, support to pervasive devices. The ISP platform configuration depend on certain requirements, but the following components are normally necessary for a basic infrastructure:

- Network Access Server (NAS) and Remote Authentication Service (RAS)
- Remote Access Dial-In User Service (RADIUS)
- Domain Name Server (DNS)
- Firewall
- Tools for subscriber management
- Service applications
- Web server
- Application server
- Database
- Load balancing and caching
- Pervasive device support (optional)

We have selected IBM WebSphere Everyplace Suite (WES) to build the ISP platform core (and more). It provides components for basic network infrastructure, ISP services, and pervasive services. The ISP platform can also be implemented by using separate components from different vendors, but then additional integration work may be required. In Figure 31 on page 107, we provide an overview of the products used to implement the ISP functions.

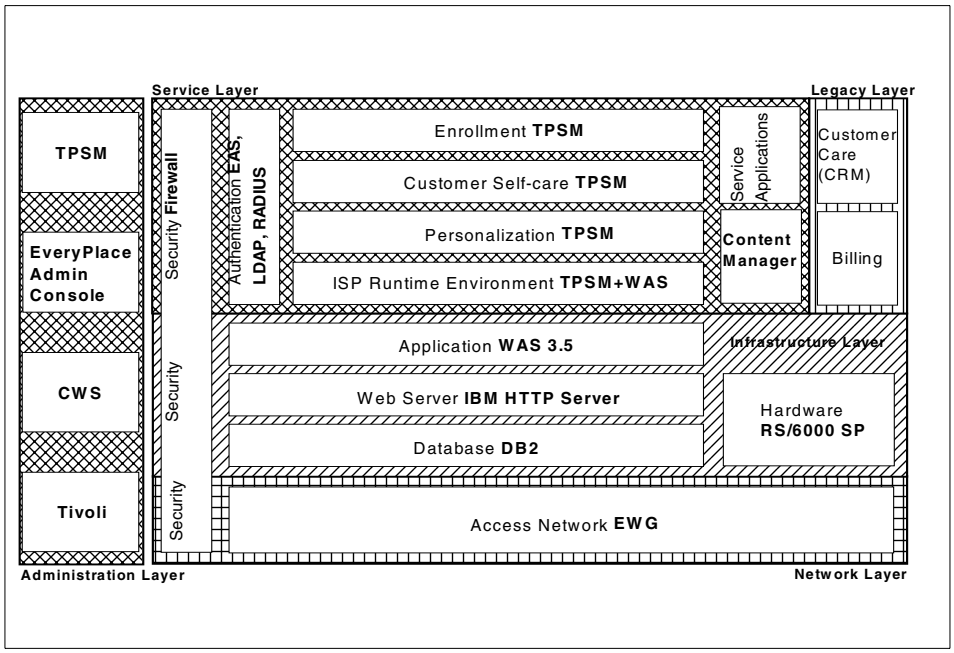


Figure 31. IBM products for an ISP

WebSphere Everyplace Suite has been designed in response to a set of fundamental business objectives that existing and prospective customers have told us they need:

- Preserve the existing investment by integrating existing applications and subsystems easily.
- Increase customer loyalty for greater retention and/or less churn.
- Deploy new services quickly and cost-effectively.
- Support new client devices quickly and cost-effectively.
- Adjust end-user experience appropriately to multiple client devices.
- Provide end-to-end security, with minimal end-user disruption.
- Provide usage information appropriate to existing accounting and billing systems.

The WES components depicted in Figure 32 on page 108 are:

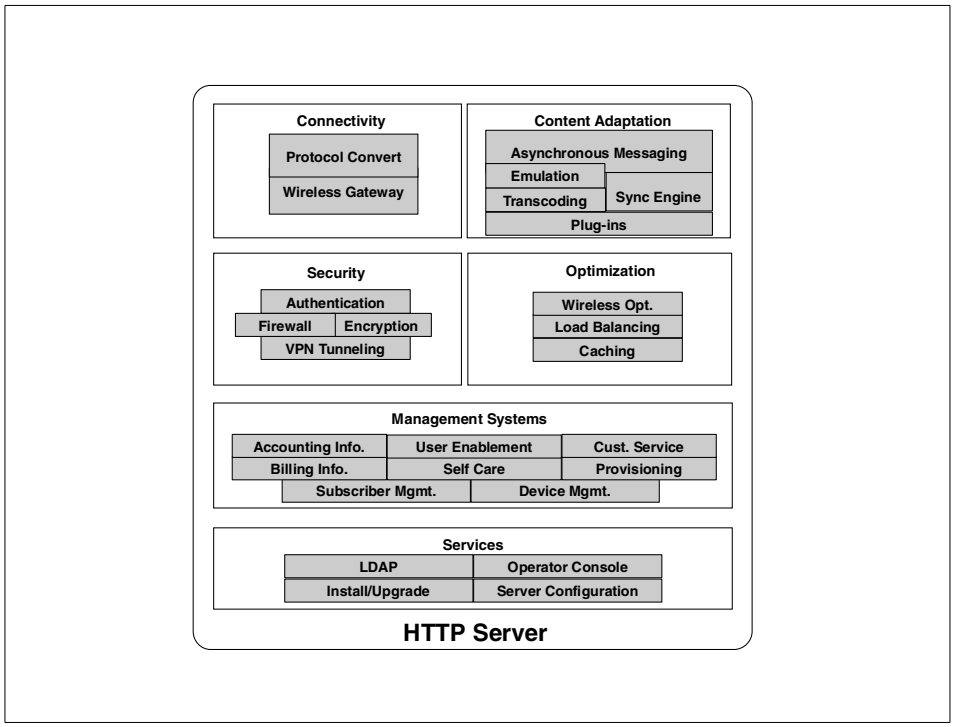


Figure 32. WES components

Connectivity

- Everyplace Wireless Gateway

  Provides secure wired and wireless connectivity between the network and the communications network, such as GSM, protocol translation, such as WAP to TCP/IP, and support for short messaging (SMS).

Content adaptation

- WebSphere Transcoding Publisher

  Transforms arbitrary content into a form that can be presented on a device that is different from the originally intended target, such as changing HTML (content intended for desktop PCs) to Wireless Markup Language (WML) (content suitable for new classes of wireless devices).

- MQSeries Everyplace

  Enables pervasive devices to queue messages and transactions, and assure their completion once and only once, in a secure and efficient manner in both connected and disconnected end user scenarios.

Management services:

- Tivoli Personalized Services Manager

  Provides a comprehensive set of management services, including content personalization, enrollment, self-care, customer care, interfaces to external billing systems, reporting, software distribution and update, and availability status.

Security

- Everyplace Authentication Server

  Provides the user and device authentication capabilities that enable a single, device-independent user log-on, and pass-through of authentication information to Web application servers.

- Everyplace Encryption

  Provides use of Two-Party Key Distribution Protocol (2PKDP). This is a secure protocol that combines bi-directional authentication with key distribution using a minimal number of messages. 2PKDP provides encryption/decryption support for all signals coming into and going out of WebSphere Everyplace Suite, with a choice of either Data Encryption Standard (DES) or RC5 encryption algorithms.

- Firewall support

  Support for integrating IBM SecureWay Firewall and popular third party firewalls, such as CheckPoint Firewall-1, to protect against unauthorized access and viruses.

- Virtual Private Network support

  Support for integrating IBM Virtual Private Network to extend an enterprise's private intranet across a public network, such as the Internet, to create a secure private connection through a private IP tunnel.

Performance optimization

- WebSphere Edge Server

  Provides highly scalable caching functions on a server to reduce bandwidth costs and improve response times when processing URLs. In addition, WebSphere Edge Server dynamically monitors and load-balances activity across the set of WebSphere Everyplace Suite processors that are deployed in a configuration.

Base (common) services

- SecureWay Directory

  A central Lightweight Directory Application Protocol (LDAP) directory which contains runtime information about active sessions, users, devices,

and networks. This database makes it easy for the various components of WebSphere Everyplace Suite (and any server that is added to the configuration) to access the runtime information centrally without having to replicate the data in other repositories.

- Everyplace Suite Console

  Provides a single console for system administrators to perform installation and diagnostic procedures, administrative procedures, and system maintenance procedures.

We first describe the base components used by WES such as DB2 or LDAP. Then we will describe each component that comprises WES.

3.2.1 IBM DB2

Information about this subject can be found at:

`http://www-4.ibm.com/software/data/db2/`

3.2.2 IBM SecureWay Directory

IBM SecureWay Directory is a Lightweight Directory Access Protocol (LDAP) directory that runs as a stand-alone daemon. It is based on a client/server model that provides client access to an LDAP server. IBM SecureWay Directory provides an easy way to maintain directory information in a central location for storage, updating, retrieval, and exchange. Figure 33 on page 111 shows what components can use an LDAP directory.

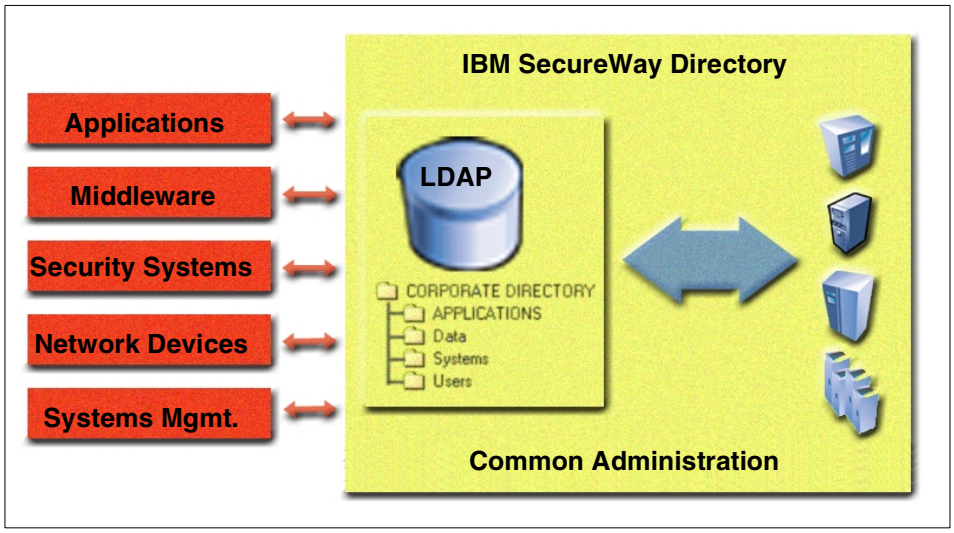


Figure 33. A single consistent view of all your directory information

IBM SecureWay Directory features overview

IBM SecureWay Directory provides the following functions:

- Interoperability with other LDAP clients
- Secure Sockets Layer (SSL) communication
- Access control
- Referrals
- Server administration utilities
- Certificate management
- Client authentication
- Replication
- LDAP directory browsing through HTTP
- Simple Authentication and Security Layer (SASL)
- Client and server plug-in support
- User/password encryption
- UTF-8 database support

IBM SecureWay Directory administration overview

IBM SecureWay Directory provides a Web-based administrator Graphical User Interface (administrator GUI) that includes online help for the administrator. From the administrator GUI, the administrator can:

- Set up the directory.
- Manage day-to-day operations of the server:
  - Start up and shut down the directory server.
  - Create, back up, and restore databases.
  - Manage Access Control Lists (ACLs).
  - Manage group membership.
  - Manage security levels such as encryption options or server certificate management.
- View or change:
  - General server settings.
  - General database/back end settings.
- Performance tuning options.
- Directory server activity.
- Directory replication configuration.

IBM SecureWay Directory security overview

The Directory Management Tool (DMT) in IBM SecureWay Directory provides a graphical user interface that enables management of information stored in directory servers. The tool can be used to:

- Connect to one or more directory servers via SSL or non-SSL connections.
- Display server properties and rebind to the server.
- List, add, edit, and delete schema attributes and object classes.
- List, add, edit, and delete directory entries.
- Modify directory entry ACLs.
- Search the directory tree.

IBM SecureWay Directory also supports LDAP referrals, as mentioned earlier in this section, allowing directory operations to be redirected to another LDAP directory server. Replication of the LDAP Directory is supported and allows for additional copies of the directory to be available for directory read

operations, which increases performance and reliability when accessing directory information.

3.2.3 IBM HTTP server

IBM WebSphere Application Server (WAS) V3.5, described in Section 3.2.4, “WebSphere Application Server (WAS)” on page 113, includes IBM HTTP Server V1.3.3 (powered by Apache). It includes the same properties as Apache Web Server, with the inclusion of some additional features.

The IBM HTTP Server supports both SSL version 2 and SSL version 3 protocols. This protocol, implemented using IBM security libraries, ensures that data transferred between a client and a server remains private. Once your server has a digital certificate, SSL-enabled browsers can communicate securely with your server using the SSL protocol. The IBM HTTP Server supports client authentication, configurable cipher specifications, and session ID caching for improving SSL performance on the AIX platforms.

The cache accelerator can improve the performance of the IBM HTTP Server when serving static pages with, for example, text and image files. Because the cache accelerator cache is automatically loaded during server operation, you are not required to list the files to be cached in your server configuration file. In addition, the server will automatically recache changed pages and remove outdated pages from the cache. The cache accelerator provides support for caching on Web servers with single and multiple TCP/IP adapters.

More information on this subject can be found at:

`http://www-4.ibm.com/software/webservers/httpservers/`

3.2.4 WebSphere Application Server (WAS)

The WebSphere Application Server (WAS) is a Java application server designed to facilitate the management and deployment of Web applications. These deployed applications are typically composed of Java Beans, EJBs, JSPs, and Servlets. They communicate to clients using a Web browser client interface. Each different type of Java application that runs in the WebSphere environment can also make use of Java Beans. WebSphere provides the environment and infrastructure required to install and manage these types of applications. WebSphere makes use of a Java development and run-time environment on the host machine. This Java environment allows WebSphere to execute the Java programs that make up Java applications. An example of the use of the WAS framework is depicted in Figure 34 on page 115.

Technology overview

The technology used in WAS is more sophisticated than traditional Java applets and Common Gateway Interfaces (CGIs). We give a short overview of the new technologies:

- Applets

  Applets are downloaded to the client browser, which generates a lot of traffic during the download. With applets, the application runs on the client machine. If these applets are too big, it is not efficient to use them in an environment where networks and devices are not capable of handling large amounts of data.

- Servlet

  A servlet is a server-based Java program used to interact with clients. Servlets are used as they run on the server. No application is downloaded on the client. This minimizes any browser dependency one could find with an applet. Finally, servlets provide a greater scalability, as the platform delivering the functionality can be scaled.

- Java Server Page (JSP)

  JSPs are HTML pages with dynamic content. This is achieved by separating presentation logic from business logic. Indeed, the page presentation is constructed statically, as a template page. This does not require extensive skills. Some or all the content of the page is determined through programming. This makes it easy for the Web page designer to create HTML pages with dynamic content.

- JavaBean

  JavaBeans are a Java component for representing and storing data in a standardized manner.

- Enterprise JavaBean (EJB)

  EJB are components that implement business logic and typically access enterprise data, transactions, and back-end applications. They are deployed in and run on EJB servers. They are distributed, persistent and transactional. More than deployed, EJBs provide a framework.

- XML

  eXtensible Markup Language (XML) is a meta-language for managing and exchanging document types and formats.

WebSphere Application Server use example

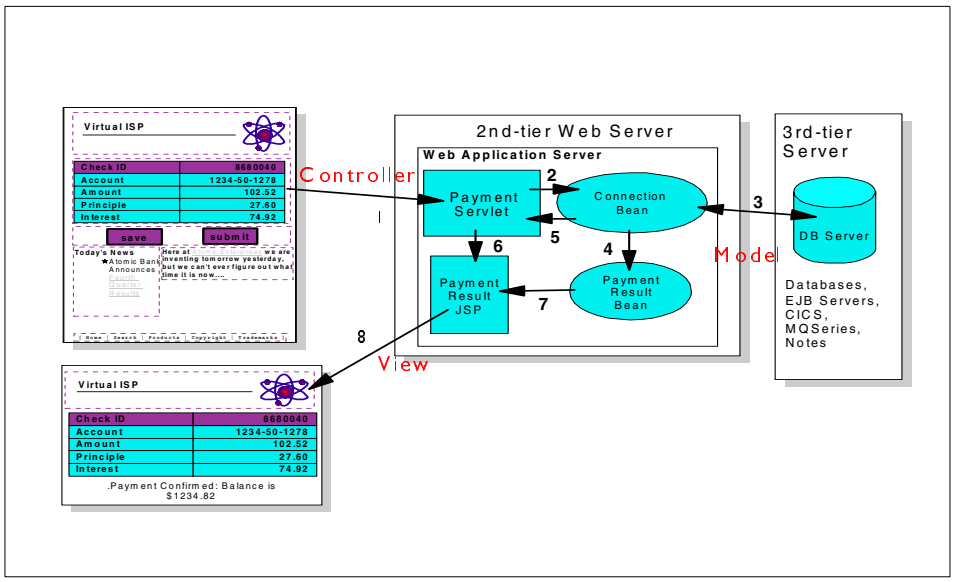


Figure 34. Flow example using WAS capabilities

In this Virtual ISP example, a customer has clicked the *Submit* button of the form:

1. The action parameter of the form calls a servlet.
2. The servlet accepts the data submitted on the form and forwards the request to the connection bean.
3. The connection bean locates data in database.
4. The connection bean creates the result bean for the user and stores the user-specific data in it.
5. The connection beans provides the return value to the servlet.
6. The servlet retrieves a JSP, which provides the template for the return page.
7. The JSP contains the result bean, which includes the user-specific data for the return page. The result bean returns data, which is placed on the return page.
8. The result page is forwarded to the customer.

WebSphere Application Server with Web servers

When an ISP runs Web applications, WAS needs to be installed on a host Web server that handles HTTP requests from browsers. WAS then delivers HTML back to them using the HTTP protocol. When WAS is installed, it modifies the configuration of its host Web server to redirect certain requests to WAS for processing rather than letting the Web server handle them.

WebSphere Application Server versions overview

WebSphere Application Server V3.5 comes in three editions: Standard, Advanced, and Enterprise. The Standard Edition includes support for JSP and servlets, as well as XML document structure services and session management. The Advanced edition has an EJB engine and database to act as a persistent store for EJB information. Transaction support is provided in Enterprise edition. The capabilities of each version is shown on Figure 35.

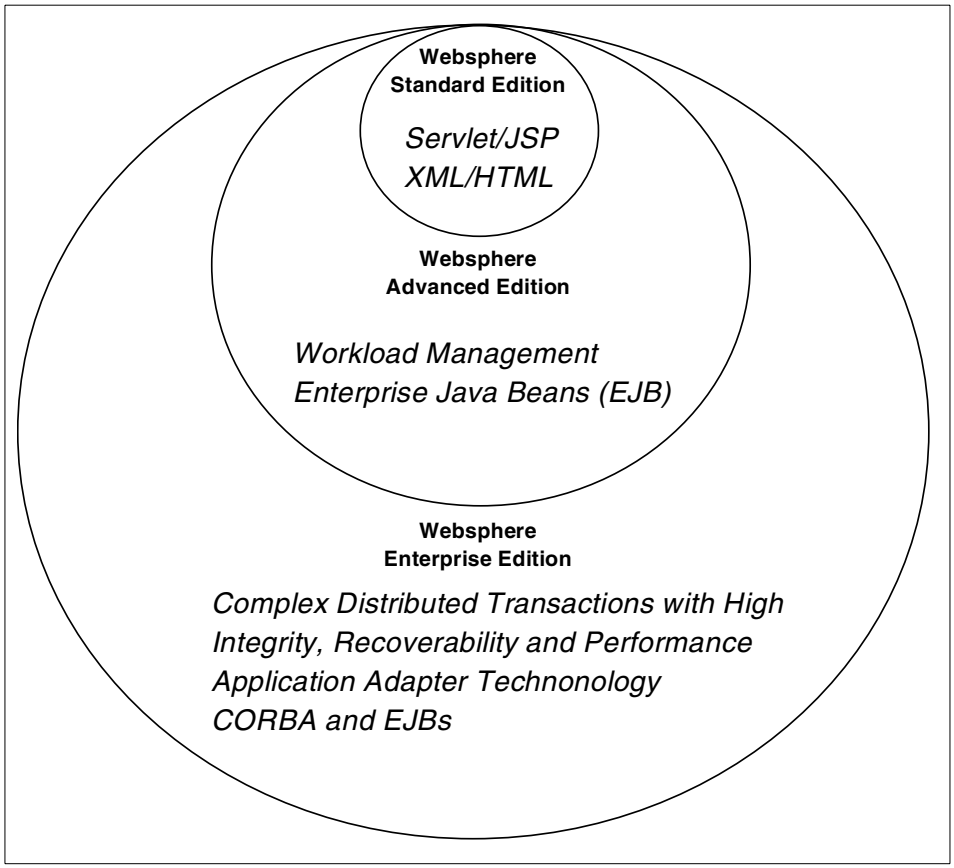


Figure 35. WAS capabilities

WebSphere Administrative Console overview

The WebSphere Administrative Console is the administrative interface of the Advanced Application Server. It can be used for a range of administrative tasks:

- Configuring resources
- Setting security policies
- Starting servers and deploying beans
- Identifying and responding to system failures
- Monitoring usage patterns

The tasks supported by the WebSphere Administrative Console fall into six categories:

- Configuration
- Operation
- Security
- Troubleshooting
- Performance
- Data storage

The WebSphere Administrative Console provides:

- A centralized hierarchical view of all resources in an administrative domain
- Guides for performing administrative operations
- Forms for viewing and modifying a resource's attributes
- A central browsing facility for JAR files
- A messages panel for monitoring critical events
- Context-sensitive help

The WebSphere Administrative Console modifies information in the repository in response to user commands and reflects any changes to the configuration and status of the administrative domain.

More information on this subject can be found at:

`http://www-4.ibm.com/software/webservers/appserv/`

3.2.5 Edge Server

Edge Server includes two components: Load Balancer and Caching Proxy.

3.2.5.1 Edge Server Load Balancer (ESLB)

ESLB is a server that is able to dynamically monitor and balance TCP servers and applications in real time. The main advantage of ESLB is that it allows heavily accessed Web sites to increase capacity, since multiple TCP servers can be dynamically linked in a single entity that appears in the network as a single logical server. This load balancing software improves the performance of servers by distributing TCP/IP session requests to different servers belonging to a group of servers. Dispatcher provides load balancing at a level of specific services, such as HTTP, FTP, SSL, NNTP, POP3, SMTP, and Telnet.

Edge Server Load Balancer features overview

ESLB creates the illusion of having just one server by grouping systems together into a cluster that behaves as a single, virtual server. The service provided is no longer tied to a specific server system. The balanced traffic among servers appears for end users to be a single, virtual server. All requests are sent to the IP address of the ESLB machine, which decides for each client request which server is the best one to accept the request. ESLB routes the clients‘ request to the selected server. The server then communicates directly with the client without any further involvement of ESLB. ESLB can also detect a failed server and route traffic around it.

3.2.5.2 Edge Server Caching Proxy (ESCP)

ESCP is used as a gateway between a client and a server for specific TCP/IP connections. Its purpose is to deliver to clients the information most frequently used that could have been stored on the ESCP machine. If the data is on the ESCP machine, it is delivered instantly to users, which minimizes the time necessary to achieve the transaction. Different algorithms can be used to decide what information is to be stored on the server.

Edge Server Caching Proxy configuration overview

ESCP can be configured in two ways:

- Proxy

  The proxy address is set up on the client machine browser. The TCP/IP destination address for all HTTP requests is the ESCP machine. Note that client browser can define exceptions to avoid connecting to the proxy for specific listed domains. The ISP platform domain should be an exception as personalized pages are no material for caching.

- Transparent proxy

  The proxy is not known by the client, but the traffic can be redirected by a router in the IP backbone of the ISP to the ESCP machine. With this configuration, the ISP is not dependent upon a configuration of the client browser.

Edge Server Caching Proxy features overview

ESCP enable subscribers to run any Web browser to access any Web server without imposing any changes to either. It significantly reduces the amount of data transmitted. ESCP client appears as a local Web proxy that is co-resident with the Web browser and communicates with it using a local TCP/IP connection and HTTP protocol.

When the browser makes a request to access information on a Web site, ESCP Client and Server is enabled to optimize exchanged information across wireless and wire line networks using intelligent caching, protocol reduction, header reduction, and data compression. ESCP also supports foreground and background queuing of browser requests and disconnected operations. It implements a technique to intercept HTTP messages that is aimed at reducing the traffic volume and optimize the communication protocol, in order to improve response time.

ESCP uses paired proxies to optimize communication in low bandwidth networks and supports an asynchronous browsing model that helps mask the effect of slow and unreliable networks. The resulting environment allows Web-based applications to be effectively used over wireless network.

More information on this subject can be found at:

`http://www-4.ibm.com/software/webservers/edgeserver/`

3.2.6 Everyplace Wireless Gateway (EWG)

The Everyplace Wireless Gateway is a highly scalable, UNIX platform which integrates data access from multiple data packet, radio, cellular, and wire line networks to enterprise LAN and WAN networks. It can therefore provide wired and wireless access directly to businesses intranets or, in our case, to the ISP platform Web servers and the Internet. EWG can also encrypt, compress, and minimize the data that passes through the wireless link, thereby increasing the speed of messaging and increasing the security of transmissions. EWG provides a set of management tools that makes maintenance easier for administrators.

EveryPlace Wireless Gateway connectivity overview

In addition to supporting TCP/IP to TCP/IP connections, the Everyplace Wireless Gateway also supports protocol translation between UDP/WSP (WAP) and TCP/IP, thereby enabling the extension of existing LAN/WAN environments to WAP compliant devices. Therefore, EWG allows the connection of all types of devices to the ISP platform and the Internet, as shown in Figure 36.

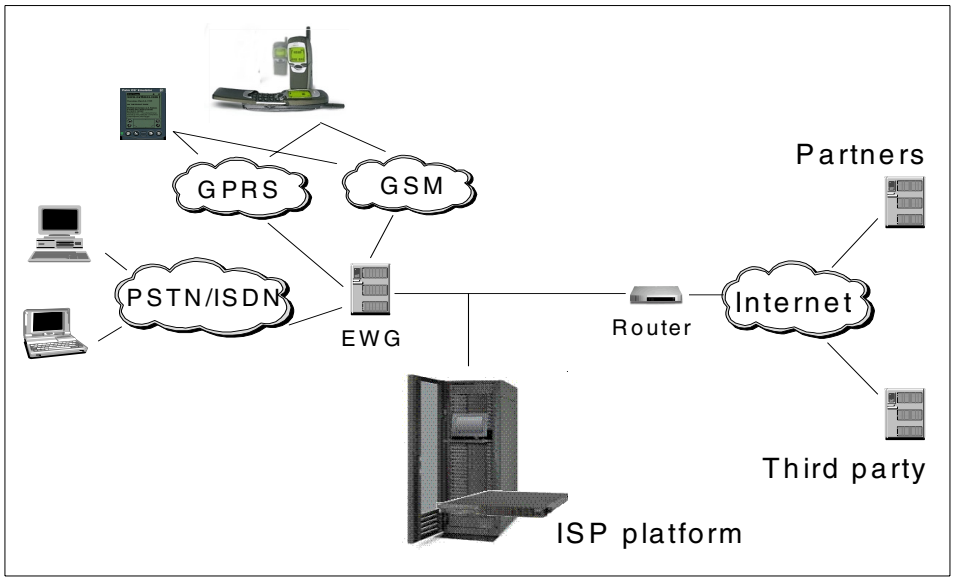


Figure 36. EWG connectivity

EWG integrates all supported networks within a single UNIX gateway host. The gateway can connect radio networks to any wire line network from Local Area Networks (LANs) to Wide Area Networks (WANs). This means that all mobile and stationary units can be linked to the same wireless gateway, regardless of the radio network, and all units can access the same set of applications. Users with different application needs, based on transmission costs, coverage, or devices, can select the best radio network for their situation. Every application using TCP/IP and WAP may communicate over EWG.

EveryPlace Wireless Gateway security overview

EWG also provides use of a secure protocol that combines bi-directional authentication with key distribution using a minimal number of messages. This protocol is called the Two-Party Key Distribution Protocol (2PKDP) and is an example of advanced security technology from IBM. When a client is

configured by the EWG administrator, it is assigned a password. This password is stored securely on EWG and is communicated (by phone, secure e-mail, or mail) to the client, so that both parties know the password. In 2PKDP, both the client and EWG authenticate each other and yet never transmit the password during the authentication process. Authentication is a prerequisite for the communication between EWG and client to be encrypted.

Data encryption helps to prevent inappropriate access to the data exchanged between EWG and clients by transforming it into an unintelligible form using the session key exchanged during the authentication process. The original data can only be decrypted by someone who possesses the session key.

Everyplace Wireless Gateway supports two different commonly used encryption algorithms, DES and RC5. When the client software is installed on a workstation, the administrator must choose the type of encryption that is desired, either DES, RC5, or no encryption. If data encryption is enabled on the client, then all IP packets flowing between the gateway and client will be encrypted with either DES or RC5 using a 56-bit key. This is shown on Figure 37.

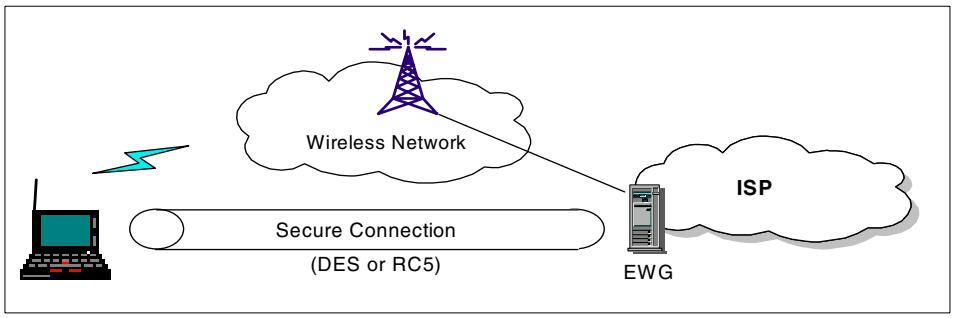


Figure 37. Secure connection over the wireless network

There are three security options to help protect your network, applications, and data:

- Client validation

  This option determines what validation is required when a wireless client initiates a session with EWG. Whenever a client logs into the gateway, the gateway must have a user name to associate with that client session. This name will be used for logging and tracing. The user name can be identified in several ways: it can be entered at the client, derived from the identifier of the mobile device being used, or it can be a default value.

- Administrator access

  With the EWG Gatekeeper function described at the end of this section, the wireless resources can be organized into operational units. Types of access that each administrator has to each type of resource in each unit can also be configured. This option is independent of the device and so applies equally to all pervasive devices, including, for example, wired PDAs.

- Data traffic control

  For security purposes and to avoid transmitting packets to a wireless client which should not receive them, EWG provides packet filtering mechanisms. Packet filtering also reduces the traffic on the wireless link. Filtering mechanisms may also be used to restrict or explicitly permit selected mobile users to access specific IP address ranges.

EveryPlace Wireless Gateway features overview

EWG and wireless clients extend TCP/IP communications to mobile clients running over a wide variety of networks.

Networks and protocols include:

- Satellite network support, providing solutions for customers in areas with poor land-based network coverage.
- Advanced Radio Data Information Services (Ardis) protocol, Mobitex protocol, dataradio network, DataTAC 5000 and 6000 networks, Modacom, and Motorola PMR support.
- Use of dial-capable digital and analog networks, such as Global System for Mobile Communication (GSM), Advanced Mobile Phone Service (AMPS), Public Switched Telephone Network (PTSN), and Integrated Service Digital Network (ISDN) networks. Native Point to Point Protocol (PPP), with PAP and CHAP, is also supported over these networks.
- Use of a LAN-based network provider and all IP-based mobile devices, such as Cellular Digital Packet Data (CDPD), and General Packet Radio Service (GPRS), among others.

Wireless connectivity features Wireless Client. It is the interface for starting and stopping communication with a Wireless Gateway. Wireless Client shields network-specific details inside the interface layer and allows IP applications on a mobile computer to run over a wireless network. For example, a radio network would not require any specialized communication protocols for use by a mobile device. IP LAN support works for wired environments and for any two nodes on a network. Using the two network

nodes, you can create a secure tunnel, which functions as a virtual private network (VPN) between the nodes.

Additionally, we find SMS has an information push capability up to 160 characters at a time to client devices using WAP protocol.

Administration is eased because of:

- EWG Gatekeeper, a user-friendly graphical administration utility, which remotely manages any number of EWGs from a variety of platforms. EWG Gatekeeper is a Java-based administration tool for EWG and wireless resources. It enables an administrator to configure wireless and WAP gateways, add users and mobile devices, define and group wireless resources, and assign administrators to wireless resources.
- Distributed gateway design, which allows multiple gateways to access configuration information from a central, LDAP-based repository.

Performance is guaranteed by fully exploiting RS/6000 Symmetric Multi-Processing (SMP), compressing headers on all IP packets, DNS caching, and packet loading.

The components of EWG are shown in Figure 38 on page 124.

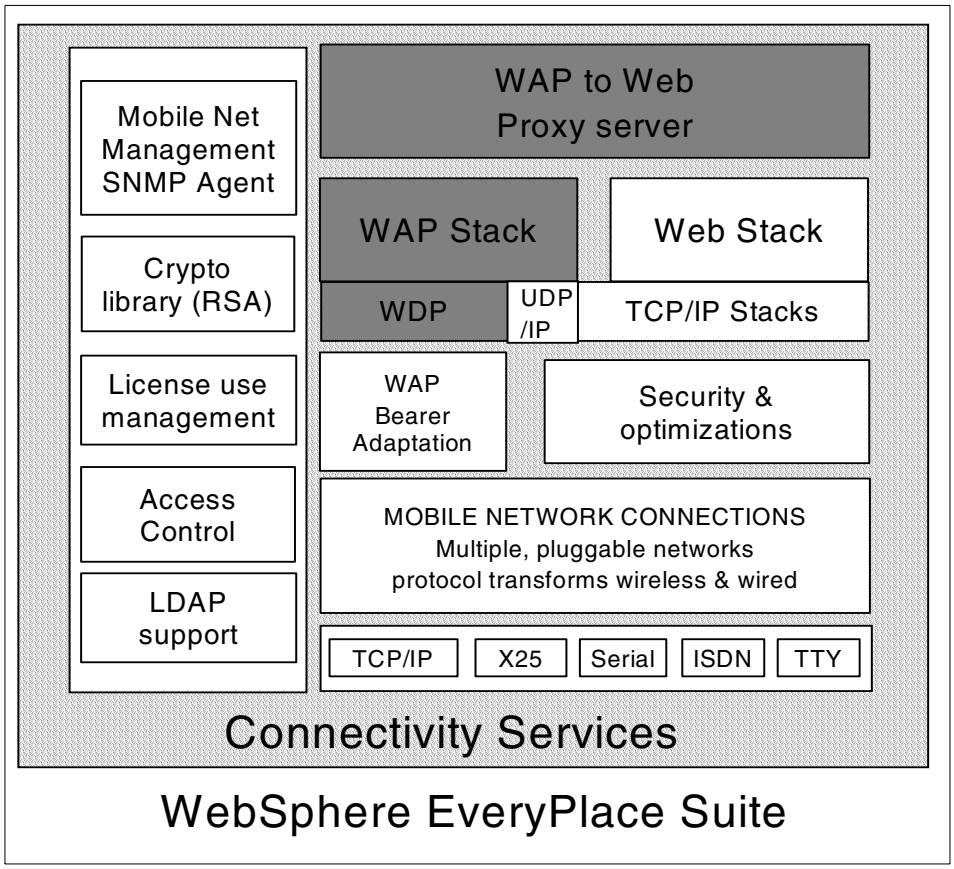


Figure 38. Connectivity services with EWG in WES

3.2.7 Everyplace Authentication Server (EAS)

EAS is the central point of user authentication for WES. It authenticates users defined to WES through the RADIUS server when they attempt to access WES services. At least one authentication server is required in the WES domain to enable integration of most WES components. It is the point of entry to the WES domain for devices that do not connect through EWG and is the next, non-firewall hop for connections through the EWG.

EveryPlace Authentication Server configuration overview

EAS runs as a plug-in to the Edge Server Caching Proxy (ESCC). ESCC is a prerequisite for EAS and must be installed on the same machine as the EAS. EAS can be configured in one of two modes that match ESCC configuration choices:

- Authentication proxy

  Performs user authentication based on HTTP Authenticate headers. In a WES domain, where the authentication proxy is installed, no other origin server, content, or application server in the WES domain may do its own user authentication. Users authenticated through the authentication proxy may not access content outside of the WES domain.

- Transparent authentication proxy

  Performs user authentication based on HTTP Proxy-Authenticate headers. In a WES domain where a transparent proxy is installed, origin servers in the WES domain may do their own user authentication. The transparent authentication proxy allows users to access material outside the WES domain.

The Authentication Server allows for single user sign-on (user ID and password) for all services within the WES domain. With this feature, user authentication only needs to done once to access services requiring a user ID and password. Authentication will still be needed for services outside the WES domain. For example, a user logs on to an enterprise site that uses WES and gives their user ID and password, which is then authenticated by EAS. If the user wants to change their password (performed by Tivoli Personalized Services Manager), they will not have to enter a user ID and password again to access this service.

3.2.8 Tivoli Personalized Services Manager (TPSM)

TPSM 1.1, provides an integrated solutions for service providers to manage subscribers and their pervasive devices. TPSM is comprised of two distinct components:

- Tivoli Internet Services Manager (TISM)

  Component that support all subscriber management functions

- Device Management System (DMS)

  Component that supports fat clients, such as smartphones or PDAs

Tivoli Internet Services Manager overview

TISM is a robust ISP hosting system. It provides content personalization, enrollment, self-care, authentication and access control, customer care, and external billing interface.

A standard set of enrollment panels can be customized to deliver uniquely branded messages and graphics, as well as ISP specific billing plans and

payment options. Behind the scenes, a consistent array of data elements are captured from each new subscriber.

TISM is implemented as a set of services and interfaces provided around a repository based on Relational Database Management System (RDBM) technology. Applications on the platform are typically implemented as servers that are loosely coupled from the core Database Management Systems (DBMS). This offloads the database server and moves most of service specific processing closer to the end client of the services.

Application servers take many forms, based on the particular needs of the clients they serve. At the application server level, multiple server instances are deployed to handle requests. Application servers are typically front ended by a Network Dispatcher whose job is to route requests to available servers.

TISM centrally manages authentication and access control. All subscribers are authenticated against the TISM subscriber database when they log in and their subscription profile parameters determine what they can do and where they can go. The RADIUS authentication's server role is to respond to authentication and accounting request from RADIUS clients, while the Customer Care server delivers responds to the Customer Care application request to perform business transactions.

TISM has designed an interface to the subscriber database that allows Customer Service Representatives (CSR) to view or change subscriber data during customer service phone calls. All data viewed by CSRs is completely up-to-date and any change made is immediately accessible to the billing system and any other integrated systems and TISM features.

A centralized configuration tool, the Director tool, allows access to the central controls for system administration, enrollment, customization, billing plan customization, and CSR security profiling.

Device Management Services overview

TPSM Device Manager is a component extending the TPSM framework to distribute and keep track and maintain applications or data residing on pervasive devices. Device Manager is loosely coupled with the rest of TPSM to allow it for use with third party systems. Figure 39 on page 127 shows the Device Management Services in WES.

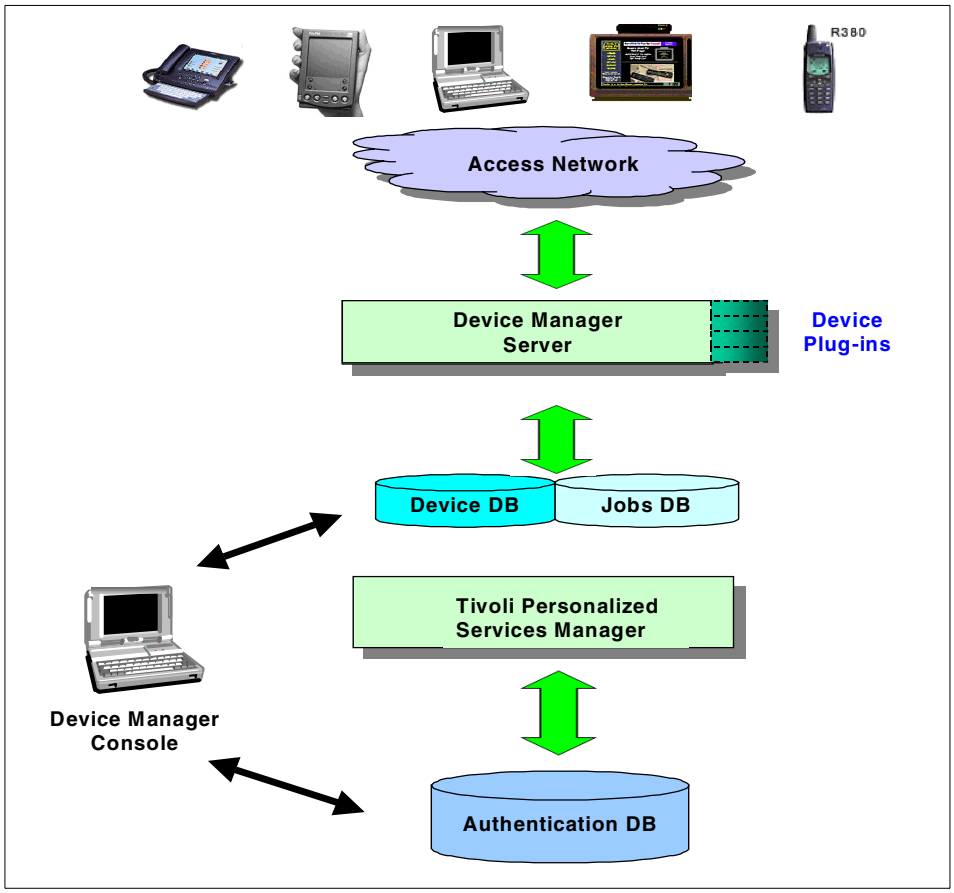


Figure 39. Device management services in WES

- Device Manager server

  The runtime components of Device Manager are implemented as Java servlets. The tasks it performs are split in two, which also reflect the architecture of the server (initial enrollment versus ongoing management of devices). A set of plug-ins are used both to interface with special protocols and to understand device specific characteristics. The set of plug-ins is extendable with new functions and with new device types. In addition to the plug-in architecture, a number of APIs exist for integration with subscriber management and other systems.

- Device agent software

  The agent allows distribution of applications or data to fat client devices. One or more agents may be needed to support devices using different operating systems or if required by other technical or business needs. Figure 40 gives an overview of this situation.

- Device databases

  The central repository is implemented in relational database storing devices and device-related data resources, including jobs scheduled for execution. The database can be shared with the subscriber management components of TPSM.

- Device Manager console

  The management console must be installed on a Windows workstation and is run as a Java Applet from a Web browser. It allows an authorized administrator or Customer Care Representative (CCR) to configure single devices, device types, jobs for distribution or configuration, and the actual software to be distributed.

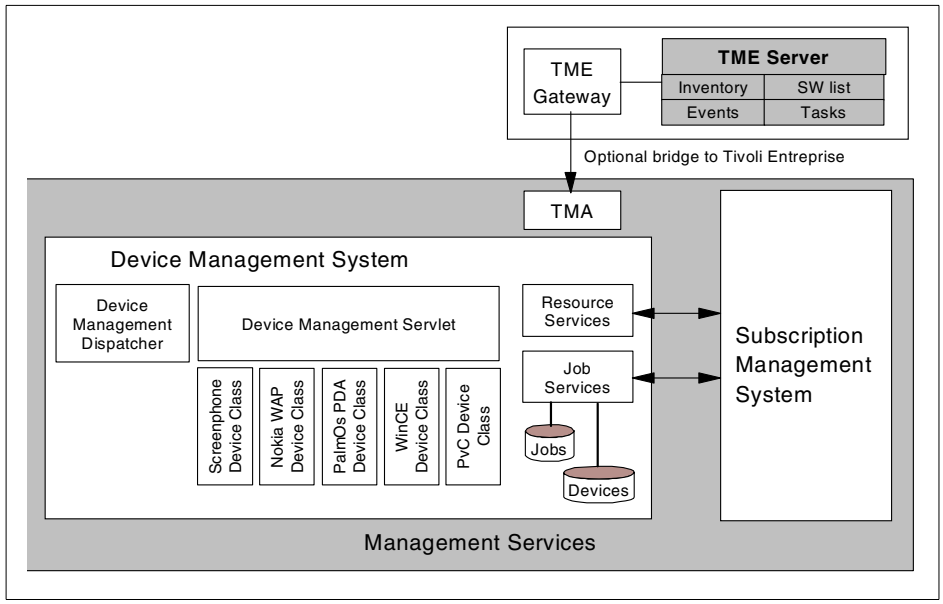


Figure 40. Device management architecture in WES

3.2.9 MQSeries Everyplace for Multiplatforms

MQSeries Everyplace for Multiplatforms is designed to satisfy the messaging needs of lightweight devices, such as:

- Sensors
- Phones
- PDAs
- Laptop computers

MQSeries provides support to mobility management and some answers to the requirements that arise from the use of unreliable communication networks. It provides the standard MQSeries Quality of Service (QoS), like:

- Once-only communication
- Assured delivery
- Messaging with other MQSeries family members
- Sophisticated security capabilities

Lightweight devices require the messaging subsystem to be frugal in its use of system resources. Consequently, MQSeries Everyplace offers tailored function and interfaces appropriate to its customer set; it does not aim to provide all the capabilities offered by other members of the family. On the other hand, it does include a number of unique capabilities in order to support its particular classes of user, such as comprehensive security provision and object messaging, together with a rich set of messaging functions.

MQSeries Everyplace architecture overview

MQSeries Everyplace messaging is carried out using queues, which are maintained within queue managers. Applications communicate with each other by connecting to a queue manager, and get or put messages to a queue. Messages that are put to a queue on a remote queue manager are sent over the network using channels, and may travel through one or more intermediate queue managers, just as in standard MQSeries.

The essentials of messaging and queuing are all present and fully supported. MQSeries Everyplace extends the scope of the messaging members of the MQSeries family:

- It expands messaging capabilities to the set of low-end devices, such as PDAs, telephones, and sensors, allowing them to participate in an MQSeries messaging network. MQSeries Everyplace offers the same once-only assured delivery and permits the two-way exchange of messages with others members of the family.
- It is designed to operate efficiently in hostile communications environments where networks are unstable, or where bandwidth is tightly

constrained. It has an efficient wire protocol and automated recovery from communication link failures.

- It supports the mobile user, allowing network connectivity points to change as devices roam. It also allows control of behavior in conditions where battery resources and networks are failing or constrained.
- It minimizes administration tasks for the user, so that the presence of MQSeries Everyplace on a device can be largely hidden, making MQSeries Everyplace a suitable base on which to build utility-style applications.
- It includes extensive authentication and encryption facilities, making it suitable for applications outside firewalls.

MQSeries Everyplace is supplied in a form more akin to a Java toolkit. As a consequence, it is a completely object oriented implementation, rather than being object oriented wrappers over a procedural base, as is the case with the existing MQSeries products.

More information on this subject can be found at:

`http://www-4.ibm.com/software/ts/mqseries/everyplace/`

3.2.10 WebSphere Transcoding Publisher (WTP)

Transcoding is the process of transforming content from one format into another, including conversion between alternative panel sizes or panel sizes and aspect ratios, so that the content can be displayed on a wide and growing variety of devices.

WebSphere Transcoding Publisher features overview
WTP offers the following features:

- A pluggable framework that hosts third-party and IBM-provided transformation plug-ins, or transcoders. New transcoders can be added and can interact with existing transcoders. All plug-ins can leverage a set of core services, such as the ability to acquire preference information in order to respond to different requests for different users or different devices.
- A base set of transcoder plug-ins that transform content. For example, one of the transcoders can select and apply the appropriate Extensible Stylesheet Language (XSL) style sheet to transcode an Extensible Markup Language (XML) document for rendering on a particular device. The framework can also host transcoders for other purposes, such as personalizing Web pages, transcoding printable documents for Web

viewing, and converting from legacy formats such as AFP (Advanced Function Presentation) to Internet formats, such as the World Wide Web Consortium (W3C) Scalable Vector Graphics (SVG). Other examples include converting HTML for display on Palm, EPOC, or WinCE devices, image re-sizing, and converting HTML to imode.

- Administrator control over configuration information and preference profiles. Administrators can also view and control messages and trace logging.
- A developer's toolkit, the IBM Transcoding Technology toolkit, is included. It contains a set of samples, instructions, documentation, and procedures to enable easy construction and implementation of customized transcoder plug-ins. Custom transcoder plug-ins can be used to process additional data formats, to support new pervasive client devices, to extract the most important elements of a particular full panel application for display on a pervasive client device, or to improve the transcoding associated with specific Web applications.

All these features are represented in Figure 41.

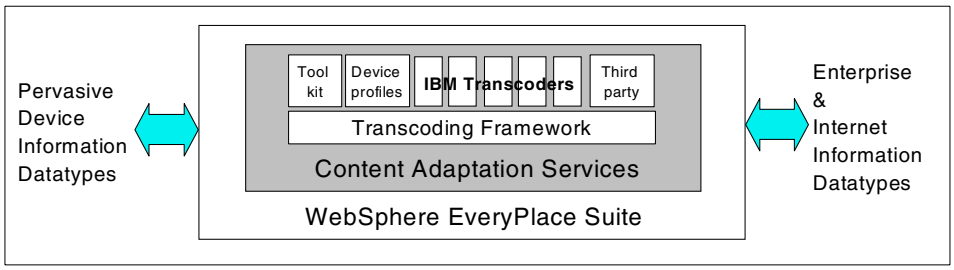


Figure 41. Content adaptation services in WES

WebSphere Transcoding Publisher architecture overview

WTP provides three distinct ways of using the transcoding mechanism on the ISP platform. These mechanisms are:

- Running as a proxy (proxy mode)
  - Standalone proxy
  - Caching proxy
  - Wireless network proxy
- Running as servlets (filtering mode)
- Running as Java Beans

WTP is generally used as a proxy within the WES solution. Other WTP configurations do provide valuable properties for the ISP platform development. We describe here all mechanisms that can be implemented using WES as a proxy and as a servlet or JavaBean:

- Running as a proxy

  The ISP can use WTP as a proxy. It is a single service that tailors content coming from many Web servers. The proxy intercepts HTTP requests and responds as they flow between users and Web servers. If users have encrypted sessions between the Web server and the browser, WTP does not make any attempt to decrypt the information that the client and the Web server exchange during the connection. It establishes a connection to the destination Web server and passes the request without looking at the data.

  If the ISP runs WTP as a network proxy but also uses a cache server or a firewall, addresses and port numbers of these servers need to be supplied to WTP.

  - Standalone proxy

    The ISP can provide the WTP as a proxy service to its users. The data that flows from the original source will be transcoded in the proxy according to the device and network profile. This method would be suitable for service providers and for those who already have Web pages and want to reach new users with new types of browser.

  - Caching proxy

    If the ISP uses an external cache server in the network, WTP can use it to store and retrieve transcoded Web pages and intermediate results. This enables WTP to avoid repeating the transcoding of frequently accessed pages, thus providing better performance. WTP and the external caching server must be within the boundaries of the same firewall. Expiration of cached data is done upon HTTP specifications.

  - Wireless network proxy

    The ISP can use WTP with the same configuration as the standalone or caching proxy. Only a wireless network profile needs to be used instead of the other network profiles. These profiles are available from standard WTP installation. The administrator needs to use the appropriate port number on the client machine.

- Running as Servlets (WAS Filtering)

  WTP can be configured to operate as a servlet, so that it can be administratively incorporated into the WAS to transcode the content produced by other servlets. The advantage of this configuration is that the

transcoding servlet can operate within the security context of the WAS, so that it can transcode information that will later be encrypted before it is sent to the client.

- Running as JavaBeans

  It is also possible to separate the transcoders from the framework and run them independently as JavaBeans. This provides means for other server programs, such as servlets, independent content-providing programs, or JSPs, to invoke single transcoders directly.

WTP Administration console

WTP provides the administration console for WTP management and configuration. The administration console in WTP allows an administrator to do the following administrative services:

- Start, stop, and restart the WTP service.
- Register new preference profiles, transcoders and XML style sheets.
- View messages, such as log entries.
- Configure and view trace information.
- Configure the framework and profiles via configuration programs with graphical user interfaces (GUIs).

More information can be found in *IBM Websphere Transcoding Publisher V1.1: Extending Web Applications to the Pervasive World*, SG24-5965.

3.2.11 Everyplace Administration Console

Everyplace Administration Console provides a centralized location where you can launch the corresponding administration console of any installed Everyplace Suite component.

More information for Everyplace Suite can be found in *An Introduction to IBM WebSphere Everyplace Suite Version 1.1*, SG24-5995.

3.3 Residential market

We present here a list of products that can be used by an ISP to provide extended services for residential users. All these services can also be offered to business consumers. The segmentation of the products is based on the type of service delivered, but any of the residential products can fit in a business environment.

3.3.1 Mail systems and unified messaging

The capability of delivering a mail service has always been the prime requisite for ISPs. This is the basic service in the ISP platform. We present here a number of mail servers. We describe their features. Each of these servers are scalable to millions. This is why we chose these products. Even if ISPs can start small, they need to choose the right product from the start. Migrating hundreds of thousand mailboxes to a new software while in production can be very difficult and hazardous.

3.3.2 Mail system - Software.com InterMail KX

Intermail KX features overview

InterMail's unique features and benefits for ISPs include:

- Class of service offerings

  Different types of mailbox accounts within a single InterMail architecture allow ISPs to offer differentiated mail services to consumer and business customers. ISPs can increase revenues through strategically customized services for core customer groups while optimizing infrastructure investments to leverage economies of scale.

- Capacity

  InterMail is a fully distributed system, scalable to millions of user accounts. As user base, message traffic or access patterns increase, resources can be seamlessly introduced to support additional loads.

- Performance

  A distributed, multi-threaded design with independent component scaling allows each server to efficiently handle many simultaneous message deliveries and requests, resulting in consistently high message throughput, even during the most demanding peak activity.

- Availability and recovery

  InterMail provides inherent support for 24 x 7 availability and rapid disaster recovery through redundant servers, failover, online backups, online maintenance, and online upgrades.

- Reliability

  InterMail inherently uses atomic transactions to prevent mail corruption and message journaling to guarantee no mail loss, insuring the e-mail service has extremely high up-time and is fully recoverable even in the event of a catastrophic system disaster.

- Spam prevention

  Curbs unsolicited e-mail, including mail blocking, relay prevention, sophisticated filtering, and easy message removal. This feature addresses a key area of ISP customer concern and reduces a growing drain on system resources.

- Internet standards support

  InterMail implements all current and emerging Internet e-mail standards: SMTP, MIME, POP3, IMAP4, SNMP, DSN, and TLS, insuring complete interoperability with all e-mail on the Internet.

- Administrative framework

  InterMail's administration framework is fully accessible through command line utilities and rich APIs that support HTML building blocks for customized management systems. Extensive logging and monitoring capabilities are provided. The system management of InterMail is easily integrated into the ISP's larger management framework.

InterMail KX architecture overview

A comprehensive high-end Internet messaging solution, InterMail is comprised of five primary components:

- Message Transport Agent (MTA) servers
- Scalable Message Store servers
- POP Client Access servers
- IMAP Client Access servers
- Directory Cache servers

Each of these InterMail components can be independently distributed across multiple platforms for flexible configuration and system scaling while offering logically centralized system management and control of the system as a whole. InterMail's distributed architecture allows deployment on a variety of platform configurations tailored to specific ISP requirements, from small-scale single-server installations to vast, multi-platform networks. Individual components can be deployed on separate or shared hardware to achieve the most efficient allocation of resources.

3.3.3 Mail system - Critical Path Inscribe Messaging Server (IMS)

IMS is a multi-threaded server that takes advantages of Operating System (OS) particular features. Its simplicity and modularity make it easy to install with recovery legacy systems like IBM HACMP. IMS server is highly scalable because it defines itself as simple redundant node in a cluster that can be

easily reassigned to new tasks, by changing its settings, without impacting existing services.

Critical Path IMS features overview

IMS is a standard compliant messaging e-mail server solution. It features:

- E/SMTP Server
- POP3 Server with APOP user password encryption
- IMAP4 Server
- A customized Webmail client

IMS also delivers security through:

- IMS authentication by:
  - SMTP user login/password
  - Dynamic list of IP addresses
- TLS/SSL implementation
- CRAM-MD5 encryption
- Anti-SPAM policy and rules:
  - Filtering message
  - Using MAPS-RBL from `http://maps.vix.com/rbl`
- Anti-Virus (achieved with Virus Detection Software: Trend Micro's InterScan Virus Wall & MIMEsweeper).

Critical Path IMS architecture overview

IMS provides a number of administration tools:

- The Inscribe Management Center (IMC)
- A GUI running on Windows NT for centralized administration and configuration.
- A simple telnet interface.
- A set of commands that perform the same operations as IMC. They are passed with a very simple telnet connection through IMS administration ports.
- A Personal Account Manager (PAM) Web customizable per-user interface in Java.
- A Group Account Manager (GAM) Web customizable administrator interface for an administrator, and only on one domain in Java.

IMC, PAM and GAM are installed on Windows NT with normal install shield installations.

PAM and GAM need the Java class swing.jar to be installed and used by the Java Virtual Machine.

IMS is installed on AIX. It is the real engine of the mail server.

3.3.4 Unified Messaging - IBM Message Center

We provide here a short overview of the capabilities of the IBM Message Center, as shown in Figure 42. A complete architectural description of the possibilities given by the Message Center is out of the scope of this redbook.

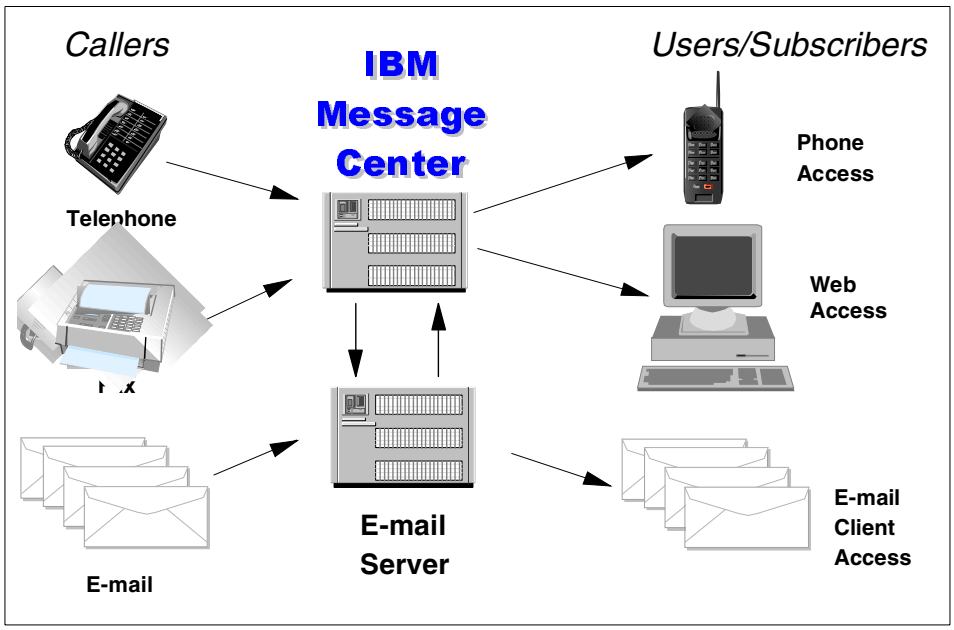


Figure 42. IBM Message Center overview

IBM Message Center features overview

- It is a scalable, carrier-grade product based on a AIX platform. Its features include:
  - 12xE1 (360) ports in a single rack server
  - 1440 ports in a single rack
  - 1000 ports per single system image
  - 500,000 mailboxes per cluster (as a lab test)

- 27 years of low cost voice storage per cluster

- Supports mobility functions.
  - Find-me/Follow-me
  - Single number for fax and voicemail
- Supports both Integrated and Unified Messaging.
  - Single and dual store architectures
- Customizable features.
  - Select or add the features required for your service.
  - Emulate existing voice-mail user interfaces.
  - Choose your "voice."
  - Choice of text to speech technologies for reading e-mail.
  - Integral Voice Response capability.
- Standards based.
  - Integrates with nearly all networks and e-mail systems.
    - AMIS

      Standard protocol that allows one voice mail server to forward voice mail messages to another server over telephone line or X.400 network.
    - VPIM

      Standard protocol allowing one voice mail messages to another over the Internet or intranet.
    - IMAP4, LDAP, SMTP/MIME
- Subscriber mailbox management over Web and phone.
  - Greetings, notification and mail handling preferences
- Subscriber classes: four pre-defined classes out of the box.
  - Business local, such as voice mail, fax mail
  - Business local and remote, such as voice mail, fax mail, and e-mail
  - Remote e-mail only, such as e-mail
  - Residential, such as voice mail
- Partitioning of administrative functions for multiple user communities.
  - Departments within a company
  - Different companies

- Outsourcing

- Create unique services with DirectTalk service creation tools.
  - Graphical tools
  - Java Bean components
  - Speech recognition and text to speech
- Platform for multiple services.
  - Unified Messaging and Voice Response
- Meets service provider requirements.
  - System partitioning and subscriber classes
  - Scalability, flexibility, high availability and redundancy
- Unique architecture.
  - Operates with external messaging systems
  - Single and dual message store
- Delivers business benefits.
  - Increased subscriber billable minutes
  - New revenue streams
  - Reduced subscriber churn

3.3.5 Nokia WAP Gateway

The Nokia WAP Server 1.0 allows users with a WAP-enabled phone to access information directly from a corporate intranet or from the Internet via their mobile phone.

The WAP Server converts IP traffic to WML, an XML-based format that lets applications talk to mobile devices, transforming them into personalized computing platforms.

More information on this subject can be found at:

`http://www-3.ibm.com/pvc/tech/nokiawap.shtml`

3.3.6 Multimedia Content

It is a time of immense change for the world's media and entertainment industries and their clients and partners. These industries, large and small, are creating new infrastructures and, in the process, are being freed from the limitations of their analog equipment. This has resulted from the new

possibilities opened by the settlement of new infrastructures, which enable faster transfer rates.

ISPs are well positioned in this market, as they manage large bases of Internet users who are potential customers for this type of service. The ISP can either seal partnerships with specialized media service providers or venture into these new market opportunities.

For these companies, creating, managing, and distributing content is an opportunity to serve and retain existing customers, reach new customers, improve their operational efficiency, create new products and, in some instances, create entirely new businesses. For ISPs in particular, this is an added-value service, which is billable on usage, which can increase their margin. This is particularly true for the residential consumer market where access revenue margins are very low.

3.3.6.1 Electronic Media Management System (EMMS)

EMMS is comprised of five distinct components, as shown on Figure 43.

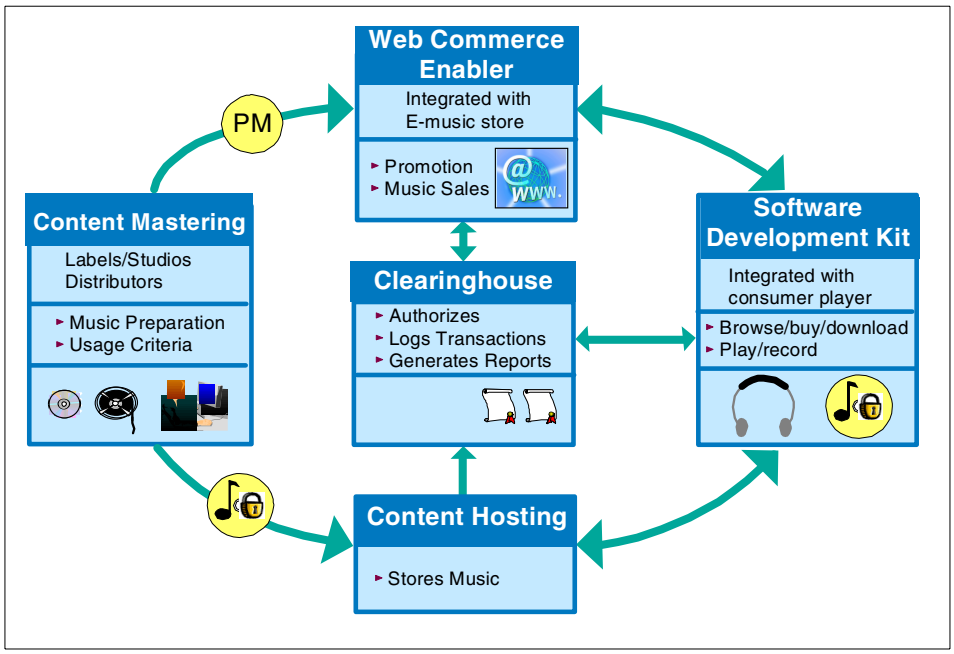


Figure 43. EMMS components

EMMS Content Mastering overview
Mastering tools to prepare and package electronic media into secure containers for transmission:

- Inputs music, metadata, and art work.
- Compresses, encrypts, packages, watermarks, and disperses content.
- Enables business model definitions based on usage criteria defined by content owners.

EMMS Web Commerce Enabler overview
Enables e-stores to make music content available for highly secure electronic download to its customers:

- Enables receipt of promotional content and creation of retail offerings, based on rights granted in the mastering process.
- Provides customer service functions, enabling the store to track electronic content downloads.

EMMS Clearing House overview
Provides rights management functions and, optionally, financial clearing:

- Verifies the licensing transaction.
- Enables customers to unlock content.
- Provides transaction reporting for royalty payment.

EMMS Content Hosting overview
Provides storage and distribution for content secure containers:

- Verifies requests for transmission of content.
- Logs and reports all transmissions.

EMMS Player/Software Development Kit overview
Allows the consumer to download music, play it, and manage a music library in a highly secure environment. Controls the permitted interaction with Compact Disc (CD) devices and SDMI-compliant devices. Allows the consumer to print art and liner notes and link to artist-related Web sites.

EMMS features overview

- Separation of trust relationships among Content Owners, Clearinghouse and Distribution System.
- Security system protects content on various forms of transmission.
- Separation of the secure distribution of content from the control of its unlocking and use.

- Content managed on an individual consumer basis.
- Upgradeable architecture with plug-ins that enable the system to evolve over time with new technology.

3.3.6.2 IBM Content Manager Video Charger (CMVC)

CMVC is a solution for the delivery of continuous time media, audio and/or video, to Internet, intranet or extranet connected clients.

The video is streamed (delivered in real-time) and does not require that the file be downloaded or saved before being played by the client software. The video is pushed by the server over the network to the client. The push architecture is similar to a broadcast environment where a video stream is started by a play command and will continue until stopped.

IBM Content Manager Video Charger features overview

CMVC features:

- Scalable servers

  CMVC on NT offers a simple single box solution with easy installation and configuration. While not as scalable as VideoCharger on the AIX platform, it offers all the same functional capabilities on AIX. In addition, VideoCharger on NT offers embedded encoder support, allowing for real time encoding with IP multicast.

- Support for a wide range of media formats

  CMVC offers a solution for both Internet and intranet users.

  - For home Internet users, who are typically connected via slower network connections, CMVC supports the delivery of Low Bit Rate (LBR) video. The LBR video is based on the industry standard H.263 video and G.723 audio from the video conferencing industry. This technology allows audio and video to be served to home Internet users connected with 28.8 Kb modems. Using about 16 Kb/s, the LBR video will offer 8 KHz 16 PCM audio and 160X120 video at 7.5 frames per second. The LBR video can be encoded at higher quality rates to provide higher resolution or more frames per second for those clients which are connected via ISDN modems, cable modems, or an intranet network.
  - For an intranet environment, VideoCharger provides support for higher quality and higher bit rate videos. In this environment, both MPEG-1 and MPEG-2 content can be supported at rates up to 8 Mb/s. Near CD quality single-channel audio is supported at 24KHz. This flexible

support for higher quality video allows a multitude of applications to be enhanced with video in the intranet environment.

- Support for industry standard protocols

  In addition to supporting industry standard file formats for audio and video, CMVC supports the popular Internet protocols, including IP and HTTP. This allows the product to be used with industry standard applications like HTML Web browsers. It also allows the product to be used on a wide variety of network types, including LANs such as Ethernet, Token Ring, and Fibre Distributed Data Interface (FDDI), WANs, such as T1, E1, T3, E3, and Asynchronous Transfer Mode (ATM).

- IP Multicast and real-time encoding

  The IP multicast feature, in particular, allows CMVC to be used as a broadcast type server in the Internet environment. This allows a single audio or video stream to be sent to multiple people, reducing the bandwidth requirements on the network. In addition, CMVC on Windows NT offers embedded encoder support. This allows an MPEG encoder to be installed on the server and have VideoCharger directly control the encoder for functions, such as real-time IP Multicast and real-time IP Multicast with live recording of the same stream. This is a very efficient yet powerful method of providing a broadcast of a live event while recording it for later re-broadcast with minimal network load.

- Multimedia File System

  At the heart of the CMVC for AIX is a higher performance multimedia file system. The file system incorporates techniques like wide striping, real-time disk scheduling, large block-size transfers, data replication, and automatic disk calibration, to provide reliable delivery of audio and video data.

3.4 Business market

We present here a list of products that can be used by an ISP to provide extended services for business users. All the services already described in the previous chapter suit most businesses requirements. We present here products that have features that can answer specific requirements for businesses. Any of these products could also be used for residential users, depending on the ISP strategy and offering.

3.4.1 IBM MQ Series

The IBM MQSeries family provides an open scalable, industrial-strength messaging and information infrastructure, enabling enterprises and their

partners to integrate business processes. Different hardware and software platforms behave as if they were designed to work together, so you are no longer constrained by their incompatibilities. The operating methods of business can be captured and turned into intelligent rules that route the information to its target. Lastly, business process flows can be defined, visualized, modified, and automated, leading to significant quality and productivity improvements. Processes may involve application to application flows, or applications, individuals, and workgroups.

MQSeries Messaging

MQSeries Messaging provides robust middleware that integrates applications across IBM and non-IBM platforms, taking care of network interfaces, assuring delivery, dealing with communication protocols, dynamically distributing workload across available resources, and handling recovery after system problems. Programs communicate using the MQSeries API, a high-level program interface that shields programmers from the complexities of different operating systems and underlying networks. The focus is on business logic. MQSeries manages connections to the computer systems.

MQSeries Integrator

MQSeries Integrator is message brokering that centralizes the operating rules of your enterprise and performs intelligent message routing and dynamic message-content transformation and formatting, as shown in Figure 44 on page 145. MQSeries Integrator is an "intelligent router," because it exceeds the capabilities of earlier message routing programs: MQSeries Integrator knows how and where to deliver its messages. It frees up customer applications and system resources by taking over the burden of knowing how to deal with each program or platform involved in messaging. It allows applications to link so that the flow of data between them matches business processes.

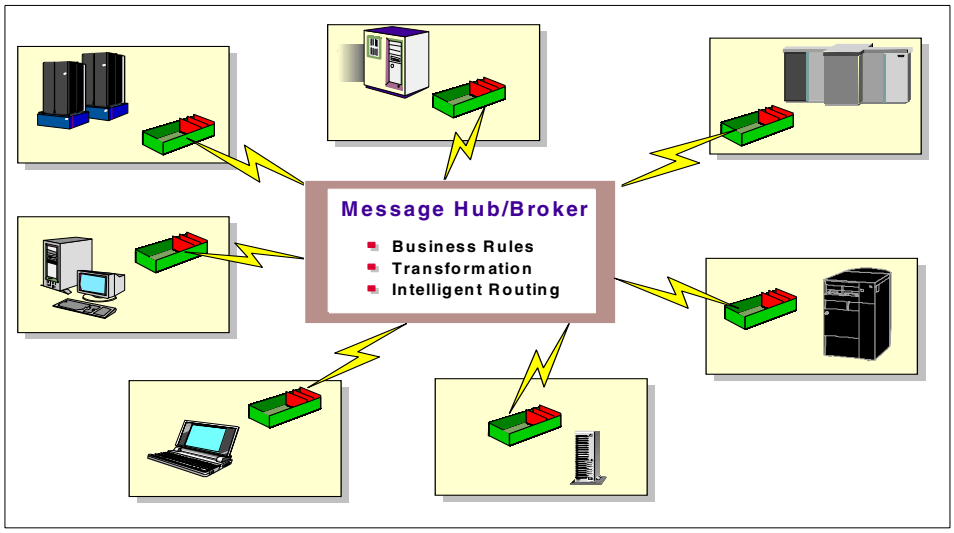


Figure 44. MQSeries Integrator

MQSeries Workflow

MQSeries Workflow enables the capture, visualization, modification, and automation of business processes, enabling application to application processes, as well as those involving people and applications.

3.4.2 Lotus Domino and Notes

Lotus Domino and Notes form a groupware product for medium and large. Lotus Domino Server and Lotus Notes Workstation is a client/server environment that allows users or clients to communicate securely over a network, and create and access documents residing on a shared computer or server.

In a Lotus Domino environment, people can work together regardless of their software or hardware platform or their technical, organizational, or geographical boundaries. The product suite includes specialized versions of Domino that can be used to make a strong standalone Web server, mail server, news server, and directory server possible. These software components allow for several useful properties on ISP platforms:

- Full Internet standards support

  Notes support for Internet standards brings it fully into the Web client space, with support for Internet-based mail, news groups, directories, HTML and more. Users can easily access ISP hosted mail and news group accounts through Notes. Wizards walk users through the set up of

one or multiple Internet accounts. Users can then view and manage all of their e-mail conveniently from a single unified Notes inbox and also get to their favorite newsgroups on the Internet. With control over things like images and tables, users can instantly create great looking pages, in full fidelity HTML, with support for native formats, such as animated GIFs and JPEGs.

- Signature files

  Signature files provide a simple way for users to personalize their e-mail by identifying themselves and providing pertinent information regarding physical address, phone number, e-mail address, and more, on every e-mail they send. Users can be as creative as they like when creating their own custom signature file using plain text or an HTML file that incorporates content like text and graphics.

- Mail rules

  Mail rules provide automatic filtering of e-mail messages. Using intuitive dialogs, users can quickly set up mail rules to filter out important e-mail messages and automatically place them in folders or delete them based on certain criterion.

- Global mail preferences

  Mail and calendar preferences are combined and users can pre-set preferences for every message sent or calendar entry created. For example, all e-mail messages can be sent high priority or spell checked, so users do not have to set preferences on each individual message.

- Group calendar view

  The group calendar feature offers advantages for users, such as administrative assistants who are required to manage multiple executive calendars simultaneously. With Notes group calendars, users can create a calendar showing their colleagues' free and busy time in a single view. If the user has been granted access to a colleague's or manager's calendar, they can view appointment details and accept an invitation or schedule appointments on their colleague's or manager's behalf.

- Calendar printing

  Users can print their calendar in multiple hard copy formats for quick reference when they are away from their office and computer.

- Mobile directory

  Users can store their entire directory on their hard drive using only a minimal amount of space. So when users are disconnected from the

network they have instant access to all their contacts, as well as Notes type ahead and type down addressing.

- Optional PDA support

  Notes supports products that synchronize with personal digital assistants.

3.4.2.1 Lotus Domino and Lotus Notes components overview

Two primary components, Lotus Notes and Lotus Domino, compose a solution for an ISP. Both the Domino server and the Notes client have a variety of Internet capabilities.

Domino Server

The Domino Server provides services to Notes Workstation users and other Domino Servers, including storage and replication of shared databases and mail routing.

The Domino server includes a built-in Web server, a Web browser for use by Notes clients, an SMTP/MIME message transfer agent, both POP3 and IMAP mail servers, an NNTP server, and an LDAP service. Domino secures the transactions of the Web server, POP3 and IMAP servers, NNTP server, and LDAP service with SSL 3.0. Domino also includes numerous Web site building tools.

Notes Workstation

The Notes Workstation communicates with one or more Domino Servers, providing an interface that allows a Notes user to access shared databases and to read and send mail. Lotus Notes Workstation combines an application development environment, a document database, and a sophisticated messaging system, providing the power to create custom applications for improving the quality of everyday business processes in areas such as product development, customer service, sales, and account management.

At its most basic level, Lotus Notes Workstation is a document database, serving as a repository for both textual and other information, for example, images, presentations, and spreadsheets. Lotus Domino Server and Lotus Notes Workstation provide the ability to distribute this information throughout an enterprise via replication, yet only those who need to see the information have access to it. In short, the intent is to improve *Communication*, *Coordination*, and *Collaboration* across any enterprise.

Shared databases exist on Domino Servers. Users place icons representing individual databases (for example, the mail file, bulletin boards, and documentation databases) on their workstations in their individual work spaces. By selecting an icon, a user can open a database to perform such

actions as accessing an existing document or creating a new document. Users also can maintain local (non-shared) databases and replicate these databases so that users always have access to the latest version of a document. Replication is the process of synchronizing multiple copies of a database so the information is the same on multiple servers.

The Domino Web Server

The Domino Web service is a full-function Web server that supports HTTP 1.1 and HTTPS. It can deliver either standard HTML files or Notes documents to Web clients. It can receive and process incoming data either with standard CGI scripts or with standard Notes programming. When Domino delivers a Notes document to a Web client, it converts it to HTML at the time of the client's request for the document, so the document it delivers is absolutely up to date.

Domino servers can also retrieve data from and deliver it to all sorts of non-Notes data sources, such as relational databases and mainframe computers. Because Notes has a rich user interface, it is a good front end to those kinds of data sources. It also makes a good Web front end to those data sources.

Internet Mail

The Domino 4.6 server can act as a full-function SMTP/MIME message transfer agent, POP3 post office, and IMAP post office. The SMTP/MIME message transfer agent can translate between Notes and SMTP/MIME message formats when necessary. It can also just transfer messages if translation is not necessary. This would be the case if the sending and receiving users both use the same message format, either Notes or SMTP/MIME.

The Domino 4.6 server can store messages in both Notes and SMTP/MIME format. Users can read a given message with either a Notes client or a standard mail reader.

The Notes client includes a choice of built-in Web browsers, plus access to the shared server-based browser, and a POP3 mail client. Notes users can pick up mail from and deliver it to POP3 mail servers, either in addition to using their standard Notes mail or instead of it.

Directory Services

The Domino 4.6 server supports the Lightweight Directory Access Protocol (LDAP). LDAP sets forth a standard way for computers to provide and access directories over TCP/IP networks.

3.4.3 IBM Content Manager

IBM Content Manager is an offering built on existing products and enhanced to be Web-enabled. The basic concept is to manage any data, store it in its native format, and distribute it on demand. Structured and unstructured data are distinguished. A library stores the index and meta-catalog, manages the security, and supports database integrity.

IBM Content Manager is well suited when the ISP needs to deliver content to its users. From the segmentation presented in Chapter 1, “What is an Internet Service Provider environment?” on page 1, this type of requirement is found for Portal ISPs rather than Access ISPs. The ISP delivers capability to store content for customers, but it also lets customers store any type of data with the ISP. This is an ideal way to build communities within the ISPs and avoid turnover.

The functionalities of IBM Content Manager can be enhanced by add-on products:

- Web distribution

  Enterprise Information Portal (EIP), with thin-client support

- Advanced Workflow

  MQ Series workflow

- Security

  Content Manager Cryptolope services

- Integration with other Applications

  Application Program Interface (API) and Object Linking and Embedding (OLE)

- Automatic data capture

  Content Manager OnDemand

- Digital audio and video delivering

  Content Manager Video Charger, described in Section 3.3.6.2, “IBM Content Manager Video Charger (CMVC)” on page 142

- Real-time connectors to Enterprise Resource Planning (ERP)

  Content Manager CommonStore for SAP or Content Manager CommonStore for Lotus Domino

3.4.3.1 IBM Content Manager servers

Content Manager is a content and object management system. Content can be pieces of information, such as scanned images, word processing documents, computer coded data, rich media, or spreadsheets. These business objects are cataloged in the Content Manager library server, providing a common repository of indexes for easy search and retrieval. The business objects themselves may be located centrally or may be decentralized and close to the users. They are stored in Content Manager object servers. With appropriate user security, any object located anywhere on the network can be retrieved and presented to the user.

Content Manager library server

The Content Manager library server is the single point of control for the system. It:

- Holds index information.
- Receives and manages requests from clients.
- Controls security and access.
- Dispatches requests to object servers to accept new objects or to move objects to a client.
- Receives confirmation from the other servers that an action was completed, and, if not, initiates appropriate actions to ensure database integrity.

The Content Manager object servers hold digitally-stored content/objects, such as images, word processing documents, spreadsheets, voice clips, video, and Electronic Data Interchange (EDI). The system supports distributed object servers.

Security

Content Manager provides powerful access control and security. This enables control over who has access to which functions and which objects. Content Manager provides flexibility for controlling access to index classes, such as content or document types, workbaskets, and advanced workflow processes. By using ACLs, an administrator can control all levels of access to these resources, either by user or by group. For example, only certain people or departments might be allowed to read documents stored in the *Employee Data* index class, and even fewer might be allowed to modify, annotate, or store into this index class.

Storage

Content Manager manages the storage and migration of objects without user intervention. The objective is to balance a low cost of storage with the best

possible response time for users. This is usually accomplished by storing incoming documents and multimedia on magnetic storage during their active life cycle and then automatically migrating them to optical storage or tape when the retrieval frequency is much lower.

Content Manager utilizes Tivoli Storage Manager (TSM) for storage management on both the AIX and Windows NT platforms. TSM provides backup and archive services, and supports a very large number of IBM and non-IBM storage devices. Utilizing one or more IBM 3995 Optical storage libraries and 5.2 gigabyte industry-standard cartridges, the storage needs of any enterprise can be handled. About 20,000 images can be stored in a gigabyte.

TSM can also be used on the ISP platform to perform backup of all critical information as part of a disaster recovery plan.

Scalability
Scalability is critical for storage systems as your business grows or as you add new areas of your business to your system. With Content Manager's highly scalable architecture, you can grow vertically by moving to a larger library server, and horizontally by adding a larger or additional distributed object servers. The importance of scalability can not be underestimated, considering the pain and expense of converting to another system when the initial system can no longer keep up with your growth.

3.4.3.2 IBM Content Manager client
The IBM Content Manager clients enable the ISP to manage and author the content they store and make available for their customers. A specific client enables, through the Web, access to the content database with full search capability and other features.

Standard functions
Content Manager comes with a prepackaged, no-charge client that provides an easy-to-use graphical interface for Microsoft Windows 95, Windows 98, Windows 2000 and Windows NT. Most functions are simple point-and-click operations. Users can perform searches and work with documents or folders that have been routed to them. Tables of contents for folders provide the information users need to select the necessary documents or business objects for processing.

Presentation
The client displays many different content types, including image documents and more than 150 common office document formats (WordPerfect, Word, Lotus 123, and others). Authorized users can annotate images with a wide

range of markups, such as highlighting, "type on" text, post-it notes, circles, squares, lines, arrows, or easily customized stamps. Free-form text comments can also be added to a note log, which is associated with each document and folder.

Folder management
Documents can be placed into folders and folders can be placed into other folders. Documents and folders can reside in one or more folders. Each folder can have its own indexing, and users can search for documents or folders or both in a single search. The value of this multi-level folder (data) model is that it provides users with capabilities similar to paper documents and folders they may have.

The user has two approaches for viewing objects: retrieving and displaying objects in the Content Manager client, or launching native applications for access to the additional functionality of the native application. For example, with a word processing document, the word processor can be launched automatically, display the document and allow the user to edit and then re-store the document. Many companies have used this capability to add video, voice, Computer Aid Design (CAD), and other native applications to their imaging systems. In effect, the desktop and your business applications become integrated with the launching of appropriate supporting programs. The Content Manager library and object servers pull together the pointers and actual storage of the business information, in essence becoming a workgroup server for the users. All business information is in one location, allowing easy access and sharing with authorized users. In addition, the import and export functions allow users to exchange information between their other applications and Content Manager.

Searching
The basic search capability allows users to query the system and retrieve documents and folders. Many search attributes can be identified at the document level and at the folder level. An advanced search capability allows the simple creation of complex queries, which can be saved as templates for future re-use. Advanced searches help you to ensure that users can use the power of electronic searches to find the most relevant information to their queries. An automatic capability to convert dynamic SQL to static SQL queries provides for faster response with less system overhead.

Facsimile
Content Manager works with a number of industry-standard, high performance fax solutions, such as FaxPlus/Open. Incoming faxes are captured at the fax server and then automatically imported into Content Manager. Users can select documents from the Content Manager client and

then fax them out. Other facsimile solutions which accept input from a print driver can also be used for outgoing faxes.

Scanning and high-volume image capture
All Content Manager clients can capture documents with desktop scanners or by importing files directly into Content Manager. The client includes Pixel ISIS drivers, which support a variety of scanners. Once scanned, documents can be routed to an appropriate workbasket (queue) for indexing. For higher volume scanning, a batch capture subsystem is usually implemented.

Kofax Ascent Capture and Input Software's InputAccel are production-level document capture applications that integrate high-performance batch scanning, image processing, Optical Character Recognition (OCR), Intelligent Character Recognition (ICR), and document indexing into a single package that significantly lowers the cost of capturing large quantities of documents into Content Manager. These capture subsystems have release processes which initiate the batch import and index processing directly into Content Manager. They allow scanning in a disconnected mode so that you can determine the best time to upload batches of scanned documents to your Content Manager system.

Integration with business applications
In addition to the prepackaged, end-user interface, Content Manager provides integration capabilities for companies which choose to build custom interfaces tied tightly to existing business applications. The Content Manager client provides an extensive number of application programming interfaces and function calls that can be called from any programming tool supporting a C interface. The OLE automation interfaces can also be used. User exits are also available to enable integration.

Intranet and Internet access
The IBM Enterprise Information Portal (EIP) technologies complement Content Manager. It provides e-business facilitation through thin client support, advanced application development components, and full text search. The IBM EIP Client Kit for Content Manager or IBM Enterprise Information Portal V6.1, depending on user requirements, provides a wide variety of improved application programming interfaces (APIs), including Java beans, C++ Object Oriented Application Programming Interfaces (OO API), and ActiveX components for Content Manager solution development and integration.

Routing and workflow
Content Manager workflow components allow the movement of documents and folders from one work basket or queue to another. Users can simply

choose, on an ad hoc basis, where to send a document or folder. Serial workflows can also be predefined for standard work processes. Content Manager can also be integrated with MQSeries Workflow to provide a very powerful work management capability.

MQSeries Workflow and Workflow BPR provide the capability to:

- Document business process flows.
- Animate business process flows.
- Simulate business process flows.
- Automate business process flows.

Tools also help you to gather and measure the resulting production statistics, which you can then use to improve your business processes even further. Utilizing these tools, you can drive continual process improvement in your organization. Once in production, MQSeries Workflow will initiate requests to Content Manager for the retrieval or storage of supporting documents at appropriate process steps.

More information on this subject can be found at:

`http://www-4.ibm.com/software/data/cm/`

3.5 ASP market

Implementing the Application Service Provider (ASP) model is the next step after providing typical Internet services. ASP is supposed to off load the burden of managing some or all of a company's IT resources, thus enabling customers to focus on core competencies. The most important features in this area are:

- Security

  Customers expect a high level security for outsourced data. Sharing an environment with other businesses is also another concern for customers.

- Network reliability

  The network needs to be reliable to ensure a high availability of the application at any time. Integrity of the data is also guaranteed by a reliable network. Connection time outs and interruptions can be recovered using a good middleware infrastructure.

- Ability to own and manage the application being hosted

  ASPs will manage a number of applications running on many environments. Managing such infrastructure requires skilled people and adapted tools.

- Performance

  Customers expect a high performance from the use of the applications, as they were used, in most cases, to handling those locally on their computer or the application server of the company.

- Speed to market

  The success of an ASP does not only rely on delivering performance and reliable and secure applications. ASPs must deliver applications as soon as they become available. ISPs have already an infrastructure up and running, and possibly potential customers. This gives ISPs an edge for this market space.

The target market for ASPs nowadays is small and medium-size business, which can not afford to quickly build and deploy e-business applications at a low per-user cost. Keeping all these applications up-to-date can also be a long, difficult, and costly operation. In addition, such customers can be introduced to such high-level applications as Enterprise Resource Planning (ERP) at a much earlier time than normal.

The RS/6000 is an efficient platform to host applications running on AIX. The RS/6000 SP meets the requirements for managing the applications efficiently with centralized management of its nodes and the requirements for performance, providing the best power available on the market.

Where ASPs are positioned today is shown in Figure 45 on page 156. Other configurations are possible, especially with broadband access deployment, where such service could be extended to anyone.

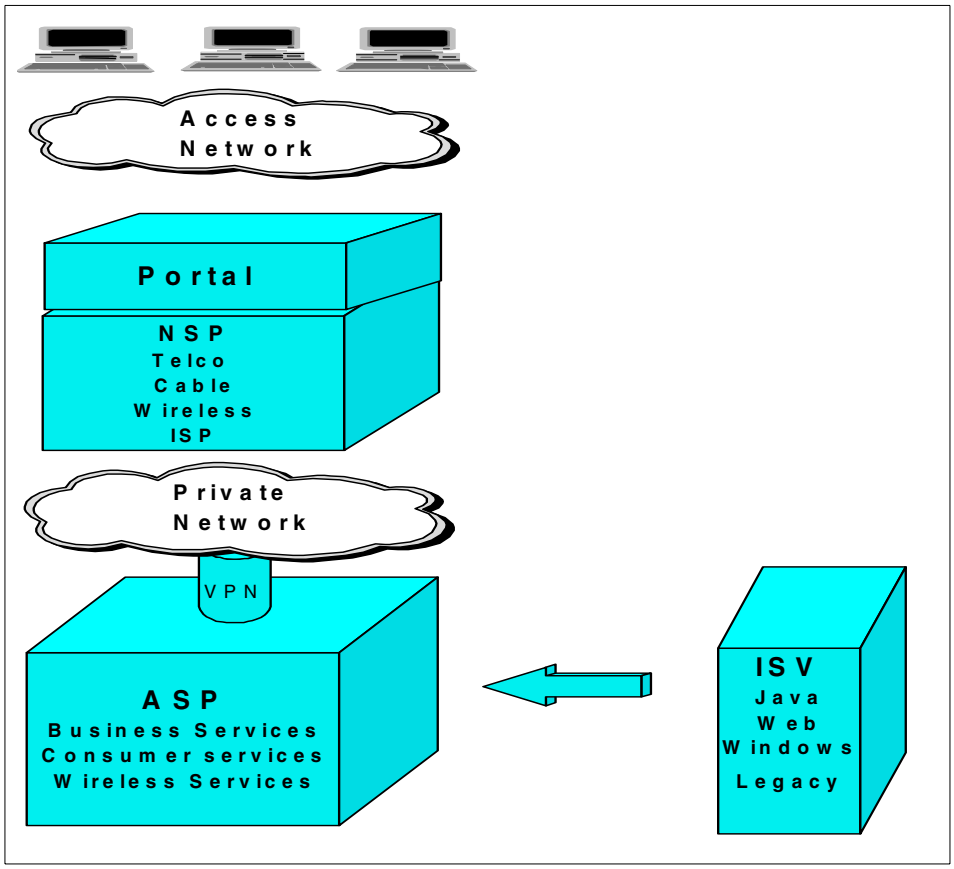


Figure 45. ASP Infrastructure

3.5.1 WAS Enterprise Edition

WebSphere Application Server has been introduced earlier in Section 3.2.4, “WebSphere Application Server (WAS)” on page 113. The Enterprise Edition provides the framework for Internet Software Vendors (ISVs) to develop their applications. We will see how WAS EE can be used in conjunction with Lotus Domino servers and Lotus Hosting Management System (LHMS) in Section 3.5.2, “Lotus ASP Solution Pack” on page 156.

3.5.2 Lotus ASP Solution Pack

Lotus ASP Solution Pack is an offering based on IBM WebSphere environment coupled with Lotus Domino environment. IBM WebSphere brings strong transactional capabilities. Lotus Domino brings communication,

coordination and collaboration capabilities. Both environments provide advanced features to host any type of application. We introduce Lotus Hosting Management System (LHMS), which is associated with these environments. The LHMS Software Developers Kit provides additional features specifically targeted to Independent Software Vendors (ISVs), which enables faster development of Web-enabled applications. Not only do they take advantage of the power of WebSphere and Domino, but they can use LHMS APIs to ease the burden of connecting the application to billing systems. LHMS is also a management tool for ASPs. ASPs are likely to manage many distinct applications in one hosting environment. The components of the architecture for Lotus ASP Pack are depicted in Figure 46 on page 158.

Lotus ASP Solution Pack features

Lotus ASP Solution Pack:

- Supports multiple applications on a single infrastructure.
- Supports multiple companies/groups on a single infrastructure.
- Supports very low touch creation of instances of applications.
- Allows a low cost addition of new applications by ASP.
- Allows the ASP to scale the solution by adding server as needed.
- Controls the cost of administration.
- Develops hooks to support different billing models.
- Tracks Licenses for Lotus and non-Lotus products.

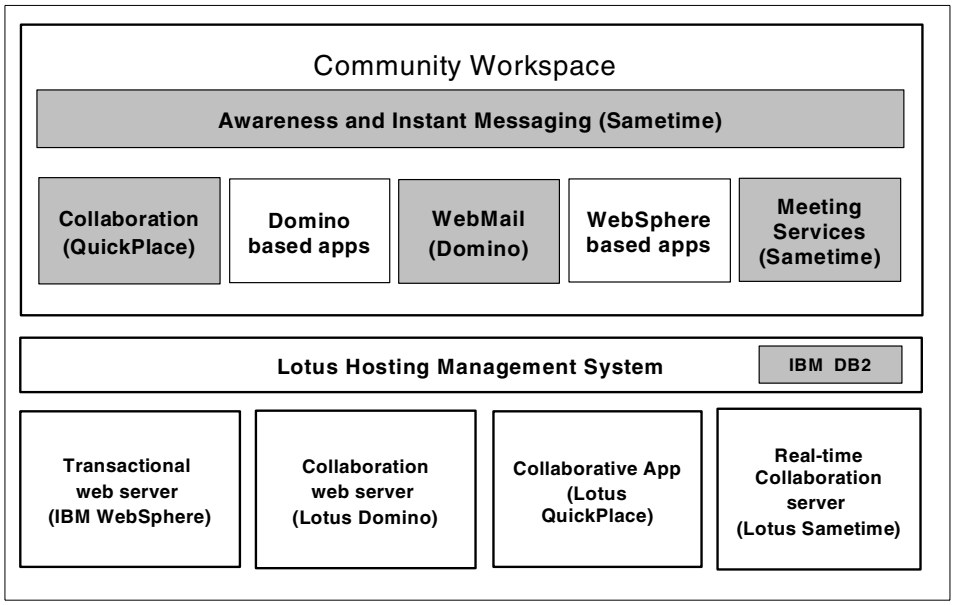


Figure 46. Lotus ASP Pack components overview

Lotus ASP Solution Pack architecture

Lotus ASP Solution Pack deployment matches what ISP infrastructures can offer. ISPs delivering services to business users will generally have strong bandwidth capabilities to deliver those services. ISPs can use this bandwidth capability to deliver more consuming applications to their customers, becoming ASPs.

In an ASP environment, the multiplicity of platforms supported has some importance. Applications can run in different environments. Rewriting the application to one specific environment is not always an option depending on the effort necessary to achieve this. On the other hand, centrally managing different environments can become a hassle. This disparity is already a part of the Lotus ASP Solution Pack offer, as the applications run on AIX and NT. The deployment of the components is shown in Figure 47 on page 159.

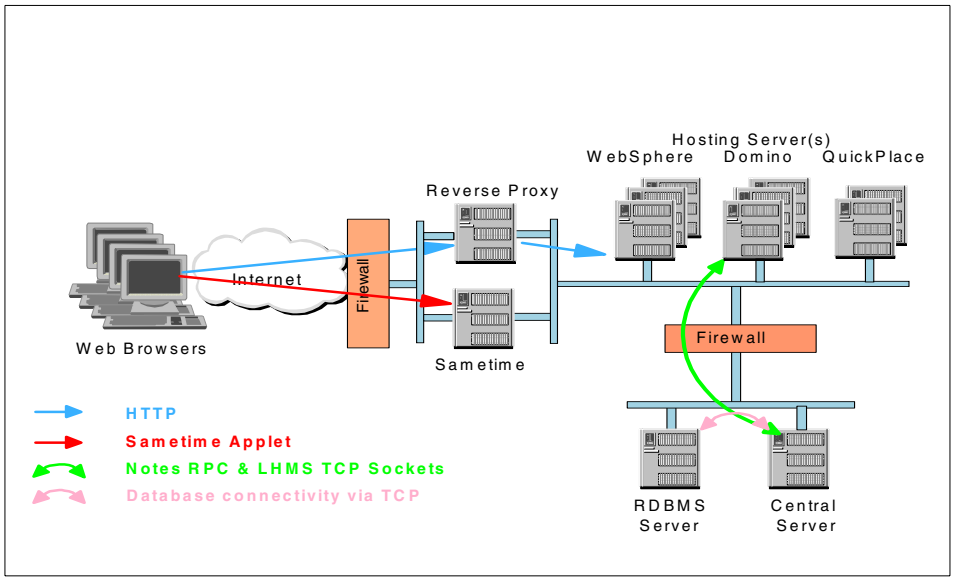


Figure 47. Deployment of Lotus ASP Solution Pack

3.5.3 Chili!Soft ASP

ASP market opportunities are only emerging, but programming skills are becoming scarce to develop Web-enabled applications. Chili! Soft is proposing Chili!Soft ASP, which uses Active Server Page (ASP) from Microsoft on UNIX IBM AIX RS/6000. ASP is an open, compile-free Web application environment that combines scripting, HTML custom server components, and robust database publishing to create dynamic Web-based business applications.

Chili!Soft ASP extends the ASP framework to major Web servers and IBM's robust AIX Operating System. ASP advantages include:

- Use of COM components. COM can be developed in:
  - C++
  - Java (thus with UNIX and NT)
  - Visual Basic
  - Delphi
  - Cobol on NT
- Data base connectivity by Active Data Objects (ADO)
- Separate development of HTML and components

- Large number of existing ASP components
- Consolidate ASP on UNIX machines
- Includes Chili!Beans
- Java to COM bridge that enables Java Objects to be incorporated as COM
- Separate the development tool of the Web Application Platform

Chili!Soft ASP architecture overview

The components of Chili!Soft ASP are:

- Five Intrinsic Objects:
  - Applications Object

    Shares information between all users of a Web Application (virtual directory).
  - Request Object

    Retrieves values passed from client browser via HTTP.
  - Response Object

    Returns information to clients via HTTP.
  - Server Object

    Provides Access to a wide variety of server utilities and information. Also provides connectivity to third party COM objects.
  - Session Object

    Enables maintenance of state, user session tracking, and storage of variables throughout a user's Web site visit.
- Object-Oriented Database Access (ADO)
- Additional Objects:
  - Included Objects allow browser capability detection and file system management
  - ASP components developed by third-parties

Chili!Soft ASP and Distributed Object technologies

- COM/DCOM

  COM (the Component Object Model) is Chili!Soft ASP's native object model on both Windows NT and UNIX. DCOM (for Distributed COM) is Microsoft's enhancement to COM that lets a COM object running on another machine appear to the calling application as if it were running locally. Chili!Soft ASP v3.0 supports DCOM. On the machine running

Chili!Soft ASP, the custom object is registered with COM as a DCOM object, with all of the information regarding the object's location (the object's name, the hostname of the remote machine, and so on). On the remote machine, the object is registered with DCOM, with security information regarding who is allowed to access the object. Any calls to this object from an ASP page are sent by COM to the remote object via RPC (Remote Procedure Calls). The remote object completes the request and sends the results back to be used in the ASP page.

With DCOM capability as part of the Chili!Soft ASP package on UNIX, developers can immediately create distributed ASP applications that use DCOM to access remote objects and systems. (DCOM is already available as part of the operating system on Windows NT.) For example, a Chili!Soft ASP application on UNIX could use DCOM to access a remote COM object running on a Windows NT system.

- Enterprise JavaBeans (EJB)

  The EJB component is first developed, then registered with the EJB Server. Then the Java RMI (Remote Method Invocation) Stub compiler is used to create stub and skeleton classes that implement the remote communications between the calling application and the remote object. When used in conjunction with Chili!Soft ASP, the stub class stays on the same machine as Chili!Soft ASP, and is registered as a COM object with the Chili!Beans cbreg tool. Chili!Soft ASP scripts are then able to create and use the stub class as they would a local object. Any requests for methods and properties of the object are passed from Chili!Soft ASP to the stub class, which is responsible for using RMI to communicate with the remote skeleton class. The remote skeleton class contains methods that dispatch calls to the actual implementation of the object running in the EJB application server.

- CORBA

  The combination of Chili!Soft ASP's new Chili!Beans technology and the Java IDL features of Sun Microsystem's new Java2 platform means that Chili!Soft ASP developers can use CORBA as one of their distributed application technologies. It also means that Chili!Soft ASP applications now have a means to connect to existing corporate CORBA services.

Active Server Page and component development

For development of ASP components for UNIX, it is often easiest to begin development of the component on NT and then port the component to UNIX. A COM component toolkit can be purchased from MainSoft (the developer of the COM implementation for UNIX used by Chili!Soft ASP) for the creation of UNIX-based ASP components. With the component source code from NT, the

COM component toolkit, and your UNIX platform's C++ compiler, you can quickly build your ASP component for any of the UNIX platforms supported by Chili!Soft ASP. The COM component toolkit then allows you to register the COM component for immediate use in your ASP pages.

Chili!Beans provides the ability to create wrapper for any Java class such that it can be treated as a COM component. An instance of any Java class can be constructed, allowing any public methods of the resulting object to be called and any of its public fields accessed, all via COM. Because of Chili!Beans, ASP developers can use any Java development tool on any Chili!Soft supported platform to create ASP components. The ASP component and the calling ASP script are totally portable, without recompiling, to any Chili!Soft ASP or Microsoft ASP installation.

Another benefit of Chili!Beans is that it enables you to use off-the-shelf Java classes and non-GUI JavaBeans in your ASP applications.

3.5.4 Citrix MetaFrame for UNIX

Citrix MetaFrame was initially a product enabling client/server connectivity over many network protocols and connections for Microsoft Windows 32 bits applications, such as Microsoft Office applications. Citrix MetaFrame for UNIX enables the same capability for UNIX and Java based applications. Citrix MetaFrame for UNIX uses the Independent Computing Architecture (ICA), which is a protocol that was created by Citrix. ICA has been already deployed on a million number scale. It is a mature technology. But ICA is, more importantly, a very efficient protocol for bandwidth use, as only a limited quantity of information is transferred over the network. This makes it an ideal solution.

Citrix MetaFrame for UNIX architecture overview

Independent Computing Architecture (ICA) is a Windows presentation services protocol from Citrix that provides the foundation for turning any client device, thin or fat, into the ultimate thin client. The ICA technology includes a server software component, a network protocol component, and a client software component.

On the server, ICA has the unique ability to separate the application’s logic from the user interface at the server and transport it to the client over standard network protocols, such as IPX, SPX, NetBEUI, TCP/IP and PPP, and over popular network connections, such as asynchronous, dial-up, ISDN, Frame Relay and ATM.

On the client, users see and work with the application's interface, but 100 percent of the application logic executes on the server. consuming less than 20 kb per second of network bandwidth.

Role of ICA
ICA is highly efficient; it allows only keystrokes, mouse clicks and panel updates to travel the network. As a result, applications consume just a fraction of the network bandwidth usually required. This efficiency enables the latest, most powerful 32-bit applications to be accessed with exceptional performance from existing PCs, Windows-based terminals, network computers, and a new generation of business and personal information appliances. With over two million ports in use worldwide, Citrix ICA is a mature, reliable technology and is fast becoming a *de facto* industry standard for server-based computing.

More information on this subject can be found at:

`http://www.citrix.com/`

3.6 Billing and CRM

The following products are utilized for billing and customer management.

3.6.1 Geneva

Geneva is a convergent billing system. It is designed to work with a range of goods and services, such as telephony services, data and content based services, or Internet services. Geneva enables billing for flat rate incomes, pre-paid services, one time events, and services measured on usage and volumes. Geneva also supports loyalty programs or products packaging.

More information on this subject can be found at:

`http://www.gtl.com`

3.6.2 Portal

Portal's Infranet software is a real-time customer management and billing solution specifically designed to accelerate the implementation of complex, multiservice Internet business models for IP telephony, wireless data, e-commerce, dial-up and broadband access services, online content and gaming, Web and application hosting, branding, e-mail and unified messaging, and other next-generation communication services.

Infranet provides the flexibility that is essential for complex Internet business models. Infranet supports the real-time creation and management of customer accounts, development, pricing and provisioning of service offerings, and activity tracking, rating and billing.

More information on this subject can be found at:

`http://www.portal.com`

3.7 Platform administration

The ISP platform is ideally managed within the RS/6000 SP environments. The RS/6000 SP provides the configuration and tools to support an ISP infrastructure.

There are ISP infrastructures that can be geographically dispersed, with a great number of machines, or simply with a highly complex nature. This is especially true when there is more than one ISP platform or when the company owning the platform has its own IT infrastructure. Efficiently managing this infrastructure is a key element in maintaining its integrity and appropriate service level.

To simplify, automate, and improve managing tasks, a management tool is necessary. Tivoli Enterprise is a software suite that best meets these requirements. It is a framework based package that enables:

- Monitoring of all distributed hardware and software, including core applications, such as DB2, MQSeries, Domino
- Event notification and processing
- Network (SNMP) management
- Remote control of all machines

The list of modules available are:

- Tivoli Management Framework
- Tivoli Distributed Monitoring
- Tivoli NetView
- Tivoli Enterprise Console
- Tivoli Remote Control
- Tivoli Module for MQSeries
- Tivoli Module for Domino

- Tivoli Module for Database

A quick overview of each product feature is provided. Note that the descriptions provided are an understatement of the capabilities of each product. Implementing such a solution requires months of effort, which is in most cases not affordable for ISPs. For large ISPs like Internet Data Centers (IDC), described in Section 1.2.7, “Internet Data Centers (IDC)” on page 11, management of a complex and distributed environment requires such tools.

3.7.1 Tivoli Management Framework

The Tivoli Management Framework makes it possible to:

- Shield administrators from platform-specific details of day-to-day operations.
- Deploy routine network maintenance with a single action.
- Deploy applications to thousands of machines with one operation maintaining availability of applications.
- Integrate with third-party applications.

3.7.2 Tivoli Distributed Monitoring

The Tivoli Distributed Monitoring makes it possible to:

- Provide continuous, predictable, and reliable user access to resources in a dynamic, highly distributed environment.
- Monitor and manage key distributed resources through a centralized management interface.
- Specify critical resource thresholds that automatically trigger customized corrective and preventive actions upon detection of conditions.
- Support standards, such as Application Management Specification or Application Response Measurement.
- Remotely access and troubleshoot distributed systems through a graphical real-time analysis tool.

3.7.3 Tivoli NetView

The Tivoli Netview makes it possible to:

- Display network topologies.
- Dynamically group any type of resources by type, location, or other common characteristics.

- Centrally set and enforce policy, in a single action, to multiple network devices.
- Graphically construct guidelines to implement current and future business policies.
- Correlate and manage events.
- Monitor network status.
- Gather performance data.

3.7.4 Tivoli Enterprise Console

The Tivoli Enterprise Console makes it possible to:

- Collect, correlate, and automatically respond to a full range of enterprise management events across any resource in any management domain, ensuring the high availability of business-critical applications.
- Centralize events and their management into one console.
- Create correlation and automation on business rules without specialized skills.
- Escalate problems that are not treated in a given time frame.
- Assign roles and responsibilities in an organization.

3.7.5 Tivoli Remote Control

The Tivoli Remote Control makes it possible to:

- Monitor, and control, PCs and servers anywhere on the network from one central location by routing the controller workstation's keyboard and mouse to a remote target workstation.
- Set policies that are centrally defined, managed, and stored.
- Determine roles by level of operations.
- Support remote users through Dynamic Host Configuration Protocol (DHCP).
- Create and access rapidly static and dynamic target lists.
- Track and manage sessions.
- Support multi-protocol networks, simplify firewall setup, and increase security.

3.7.6 Tivoli Manager for Domino

The Tivoli Manager for Domino makes it possible to:

- Simplify and centralize deployment and upgrades of Notes client software.
- Capture and correlate all events and statistics generated by Domino servers.
- Monitor performance statistics maintained by Domino.
- Centralize user administration.

3.7.7 Tivoli Manager for MQSeries

The Tivoli Manager for MQSeries makes it possible to:

- Link MQSeries management to databases, applications, and other middleware products spanning host and distributed systems.
- Deploy MQSeries to distributed platforms, with pre-distribution checks on system resources and dependencies and post-distribution validation.
- Configure the appropriate MQSeries queues and channels from a central console.
- Monitor the health of the MQSeries network across host and distributed platforms ensure its availability.
- Manage to multi-domain, cross-platform, and enterprise-scale MQSeries networks from one centralized point.
- Take appropriate responses through rules-based event correlation to MQSeries events.

3.7.8 Tivoli Database Management

The Tivoli Database Management makes it possible to:

- Manage an enterprise-wide, heterogeneous, and large database environment.
- Centrally control individual servers and databases, regardless of their location.
- Monitor all critical database performance ratios.
- Define multiple event thresholds and automatic actions for each of them.
- Apply database monitoring profiles to tens or even hundreds of databases in a single operation.
- Centrally manage policy-based definitions and synchronization of users, roles, and resources across multiple database resources.

More information on this subject can be found at:

http://www.tivoli.com/products/

3.8 Other products

Other products should be mentioned that cover ISP requirements. We did not have the time or material to describe all of them in this redbook. We will give a list of products that can be integrated into an ISP/ASP environment, depending on the requirements.

3.8.1 WebSphere family

We provide a picture of the WebSphere family of products, in Figure 48 on page 169. Most of these components could be used to help build an ISP. As the competition intensifies in this arena, requirements become more and more complex. ISP/ASP often need to diversify their activity to gain an edge over their competitors.

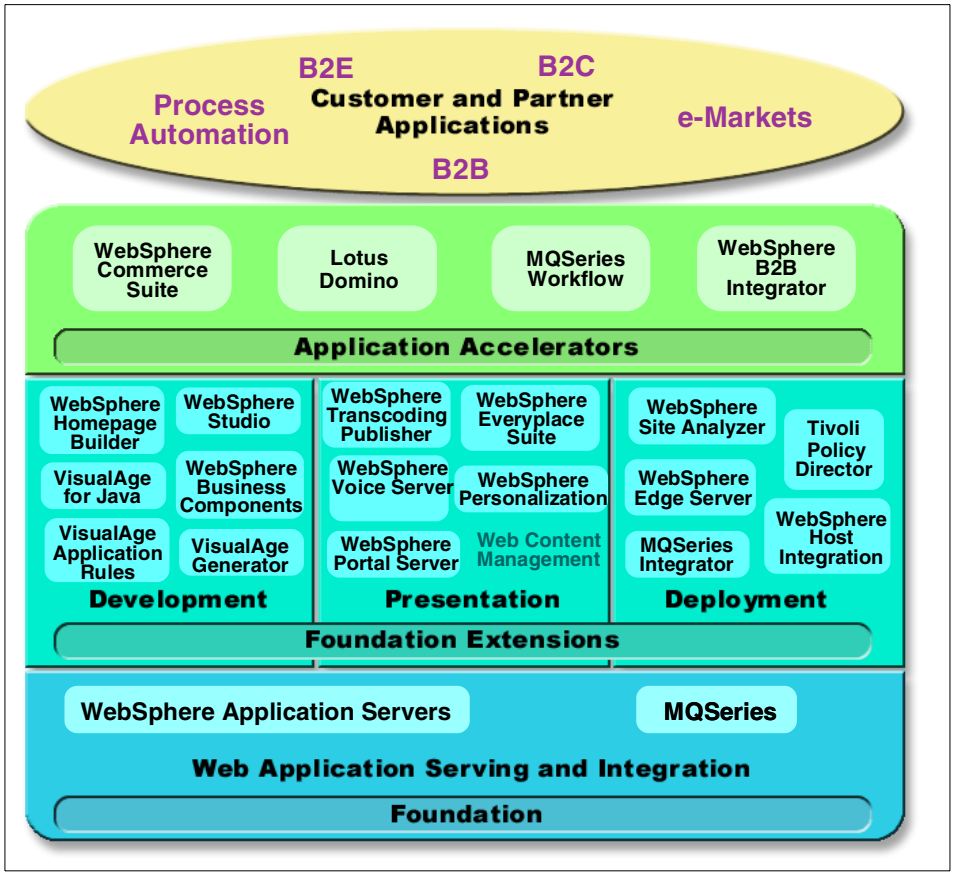


Figure 48. WebSphere Products family

3.8.2 IBM Partner products

Here we provide a list of some of our Partners' products for which we did not have the time or space to describe:

- ATG

  Offers an middleware framework with personalization and commerce server.

- BroadVision

  Offers full suite of "One-To-One" applications on AIX for next-generation integrated e-business solutions.

- DoubleClick

  Offers a full advertisement solution.

- Entrust

  Offers GetAccess application on AIX to manage secure e-business portals.

- iMediation

  Offers virtual site capabilities.

- Maconomy

  Offers business software for automating the services industries.

- RealNetworks

  Add new multimedia capabilities to Web sites.

- SendMail

  Offers a fully scalable mail solution.

- Vignette

  Offers a content management solution.

Chapter 4. RS/6000 SP fundamentals

The Scalable POWERparallel System (SP) is one of the most powerful computers in the market place. Designed for performance and scalability, this system makes the processing of applications characterized by large scale data handling and compute intensity feasible. These characteristics have extended its primary target from the scientific area to the commercial solutions world.

4.1 High-level system characteristics

The SP system is basically a group of RS/6000 boxes (nodes) working together under one point of control, the Control Workstation (CWS). It manages the whole group of RS/6000s as a unique system, from a SP with two nodes to a SP with 512 nodes.

From the product's inception, the RS/6000 is a computer system that has the following system characteristics:

- Scalability
- Use of known architectures and technologies
- Flexibility
- Manageability
- Availability

4.1.1 Scalability

The SP scales in the following aspects:

- Hardware and software components: to deliver predictable increments of performance and capacity.
- Network bandwidth: both inside and outside the machine.
- Systems Management: to preserve the investment in tools, processes and skills as the systems grows.

The SP can scale up and down and it can be subdivided, either logically or physically, into smaller SPs. In addition, scaling is a consistent process regardless of the initial size of an SP implementation. The simple ease of expanding the system also suits many customers' procurement environments. A researcher or department with some funds could easily add a node or two or upgrade the memory in an existing node, without incurring a large upgrade or swap cost.

4.1.2 Use of known architecture and technologies

4.1.2.1 Architectures

The SP does not implement special-purpose architectures. It supports the key programming models in both the technical and commercial areas. The tools and skills to program the SP are commonly known and widely available.

4.1.2.2 Technologies

The SP systems runs Advanced Interactive Executive (AIX), IBM's standards compliant implementation of UNIX, and as a result inherits a portfolio of over 10,000 off-the-shelf applications and enablers. The hardware building blocks of the system are simply RS/6000 machines. The SP attaches the same disk and tape subsystems as any RS/6000 machine, and is managed with the same enterprise-level tool, such as Tivoli and Tivoli Storage Manager (ADSM).

4.1.2.3 High-end technology flows into volume marketplace

The SP's affinity with the mainstream RS/6000 AIX marketplace allows innovations from the ultra-large SP implementations to flow into the mainstream volume marketplace.

4.1.2.4 Marketplace volumes fuel high-end development

The marketplace success of SP technology, developed to meet the requirements of the high-end customers, generates the required revenue to sustain high-end development.

4.1.3 Flexibility

The SP is a multipurpose platform addressing a broad range of customer requirements. Not only can it help solve problems across a range of industries, but a single SP can also be used for a variety of applications within a single organization. The system can be partitioned into pools of nodes. For example, two nodes can work as a Lotus Notes server, while ten others process a parallel database.

4.1.4 Manageability

Managing large systems is always a complex process. For the SP system, a single graphical operations console that displays hardware, software, job, and user status makes system management easier. The system administrator uses this console, an RS/6000 system known as the Control Workstation (CWS), and the Parallel Systems Support Programs (PSSP) software product (available with the SP system) to perform task management, including user

and password management, job accounting, and system startup/shutdown, monitoring, and partitioning.

4.1.5 Availability

The SP system optimizes high availability through built-in redundancy, subsystem recovery, component error checking and correction, RAID5, external and internal disk mirroring, and hardware and software monitoring. Clusters of up to 16 SP nodes are supported by one of the industry's leading software products for critical application backup and availability, High Availability Cluster Multi-Processing (HACMP) for AIX. If an error, such as a node failure, occurs, the system can execute a recovery script that transfers the work to another node and prevents the application from going down.

In addition, the SP system offers a wide range of open system management software tools for operations and administration, availability, deployment, and security management. Included are the Tivoli and NetView network management products, Tivoli Storage Manager for backup and recovery, and Performance Toolbox Parallel Edition for performance monitoring.

4.2 RS/6000 SP architecture

Classification of computer architectures can be done in many ways. Michael Flynn's taxonomy classifies computers based on the way streams of instructions interact with streams of data. Based on this classification, there can be four possible types of architectures, as shown in Table 1.

Table 1. Flyn's taxonomy

Flyn's taxonomy		**Number of data streams**	
		Single	Multiple
Number of instruction systems	Single	**SISD**	**SIMD**
	Multiple	**SIMD**	**MIMD**

This classification is widely accepted, and we can find practical implementations for most of these computer types. As a matter of fact, the RS/6000 SP could be classified in more than one of these categories. Modern computer architectures, in general, cross many of the lines that divide each category.

Although Single Instruction Single Data (SISD) architecture is part of what the RS/6000 SP can offer, we concentrate on the multiprocessor aspect of the system.

A Multiple Instructions Multiple Data (MIMD) machine executes multiple streams of instructions on independent executing units (processors). Within the MIMD architecture, we can further divide the classification, as shown in Figure 49, based on the way data is shared between processors. We have two categories: Shared nothing and shared data.

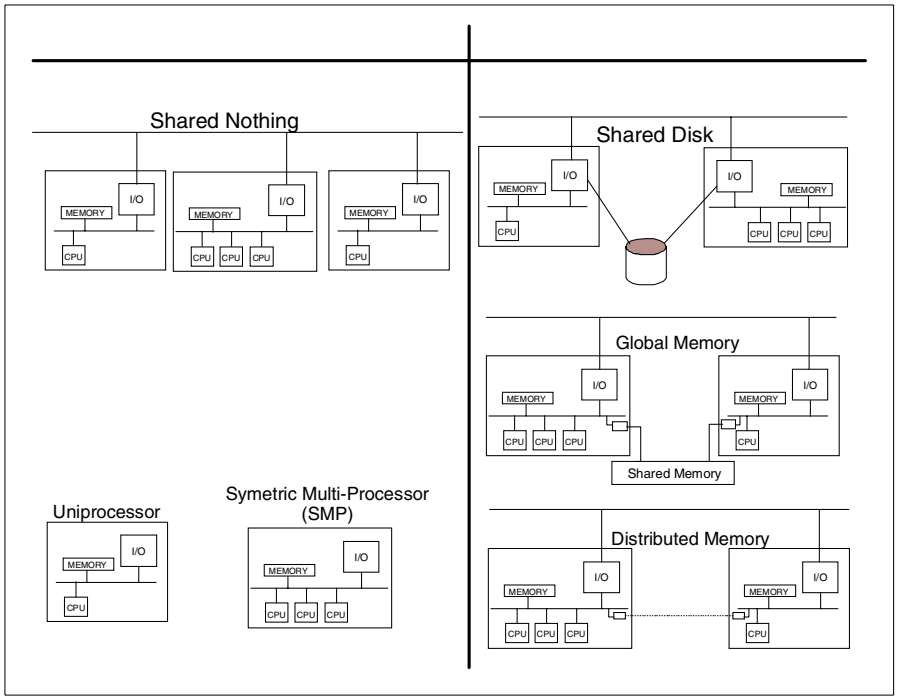


Figure 49. MIMD system's classification

Shared nothing Shared nothing is referring to data access. In this type of architecture, processors do not share a common repository for data (memory or disk), so data sharing has to be carried out through messages. Shared nothing systems are also referred to as message-based or message-passing systems.

Shared data Shared data architectures have processing units sharing data in one way or another. One of the most popular shared data multiprocessor implementations is Symmetric Multiprocessing (SMP). In this architecture, all processors share a common memory bank and I/O unit(s). All the processors, memory and I/O units are connected through a system bus (there can be multiple system buses).

The advantage of this architecture is having minimum impact to applications while taking advantage of parallel processing. Applications developed for uniprocessor systems can run on SMP machines while using a single processor. However, the system is able to handle multiple applications, which will not necessarily benefit the particular application, but will benefit the system in its entirety.

The concept of threads is associated with this architectures. Streams of instructions can be broken into several set of instructions that are relatively independent and executed in different processors. This, in theory, accelerates the application as a whole, but in reality, the effectiveness of multi-threading an application will ultimately depend on the interaction between threads and data.

Scalability is a limitation in SMP architectures. While adding processors to an SMP system may improve its performance (by adding more executing units), the shared elements within this architecture will slow down the system to a point where adding more processors will prove to be counterproductive. However, this self-imposed limitation has been overcome year after year, thus increasing the practical number of processors in an SMP machine to numbers we would not have imagined a few years ago.

If you analyze the RS/6000 SP from a system point of view, you see that processor nodes communicate with each other through communication networks (Ethernet, SP Switch, or any other supported network). So, by definition, the SP architecture is a Shared Nothing architecture. However, each node could be classified as a SISD node (old uniprocessor nodes) or a MIMD node (such as SMP nodes). Furthermore, the RS/6000 SP flexibility allows having clusters of nodes running within the RS/6000 SP umbrella.

4.2.1 The SP as a cluster

The concept of a cluster within the RS/6000 SP was initially developed as a system partition. These system partitions were intended for switch isolation and software coexistence. The main idea was to divide the SP into several functional SPs, controlled by the same control workstation, but having different software levels and system management realms.

At the core of partitioning is the System Data Repository (SDR) and the partition-sensitive subsystems, which include the RS/6000 Cluster Technology (RSCT). At a high level, applications can take advantage of the RS/6000 Cluster Technology by using the notification and coordination services provided by RSCT. At a system level, administrators can take advantage of the high availability functions provided by the basic PSSP

components, as well as the Enhanced Scalability version of the High Availability Cluster Multiprocessing (HACMP ES).

The RS/6000 SP is, by definition, a cluster. If you consider each node as an independent machine, running its own instance of the operating system, having its own address space and data access but tied together by a set of software layers, then the cluster classification comes naturally.

4.2.2 The SP as a parallel machine

The RS/6000 SP provides several facilities to develop and run parallel applications. However, an application needs to be specially designed and developed to take full advantage of the true parallelism that the RS/6000 SP can offer.

The SP Switch network provides redundant high-bandwidth low-latency paths between nodes. Applications can use these paths to share data and state information at high speeds.

Parallelism is achieved by using multiple nodes to execute a job where each task within this job can be allocated to a different node. The RS/6000 SP provides facilities for data sharing such as:

- General Parallel File System (GPFS)
- Message Passing Interface (MPI)
- LoadLeveler

These components will be explain further in this chapter.

The RS/6000 SP architecture offers parallel application scalability that can be measured in two ways: vertical and horizontal.

Vertical scalability The ability to increased the processing power of a single node. Assuming that a job will execute in multiple nodes, vertical scalability benefits the application by increasing the processing power of each node. This is the easiest way to improve performance of an application, because it does not require any changes to the application itself.

Horizontal scalability The ability to add more processing power to the application by adding additional nodes. This is the most natural way to scale a parallel application, but it may involve modification to the application to take full advantage of the additional processing nodes.

4.3 Hardware components

The basic hardware components of the RS/6000 SP system are:

- Frames
- Processor nodes (includes SP-attached servers)
- Extension nodes (includes SP Switch Routers)
- Switches
- A control workstation (A high availability option is also available.)

These components connect to your existing computer network through a local area network (LAN), making the RS/6000 SP system accessible from any network-attached workstation.

Figure 50 on page 179 illustrates a sample configuration of these hardware components. It gives you a rough idea of how they are connected. Thin nodes and the SP Switch are mounted in a tall frame. The thin nodes, SP-attached server, and the SP Switch Router are connected to the SP Switch. The thin nodes, the SP-attached server, and the SP Switch Router are connected to the SP Ethernet interface of the control workstation. The thin nodes, the SP Switch, the tall frame, and the SP-attached server are connected to RS-232C interface on the control workstation.

4.3.1 Frames

The frame is a drawer that contains the nodes, switch and extension nodes.

There are several classes of frames that can build an SP. There are a total of five options that are always compatible with all existing SP software:

- Short frames
- Short expansion frames
- Tall frames
- Tall expansion frames
- SP Switch frames

4.3.2 Processor nodes

The IBM RS/6000 SP System is scalable from 1 to 128 processor nodes, which can be contained in multiple SP frames. Up to 16 processor nodes can be mounted in a tall frame, while a short frame will hold up to eight processor nodes. SP systems consisting of more than 128 processor nodes are available.

All the nodes currently available are in the Symmetric MultiProcessor (SMP) configuration and use Peripheral Component Interconnect (PCI) architecture.

The following are the SP processor nodes:

Thin nodes Thin nodes occupy a half drawer, allowing sixteen thin nodes in a tall frame and eight in a short frame.

Wide nodes Wide nodes occupy one full drawer, allowing eight wide nodes in a tall frame and four in a short frame

High nodes High nodes occupy two drawers, allowing four high nodes in a tall frame and two in short frame.

SP-attached servers The SP-attached server is housed in its own frame and has both node-like and frame-like characteristics. Like a standard SP processor node, the SP-attached server can perform most SP processing and administration functions.

For further information about available the type of nodes and attached servers, refer to the following URL:

`http://www.ibm.com/servers/eserver/pseries/hardware/factsfeatures.html`

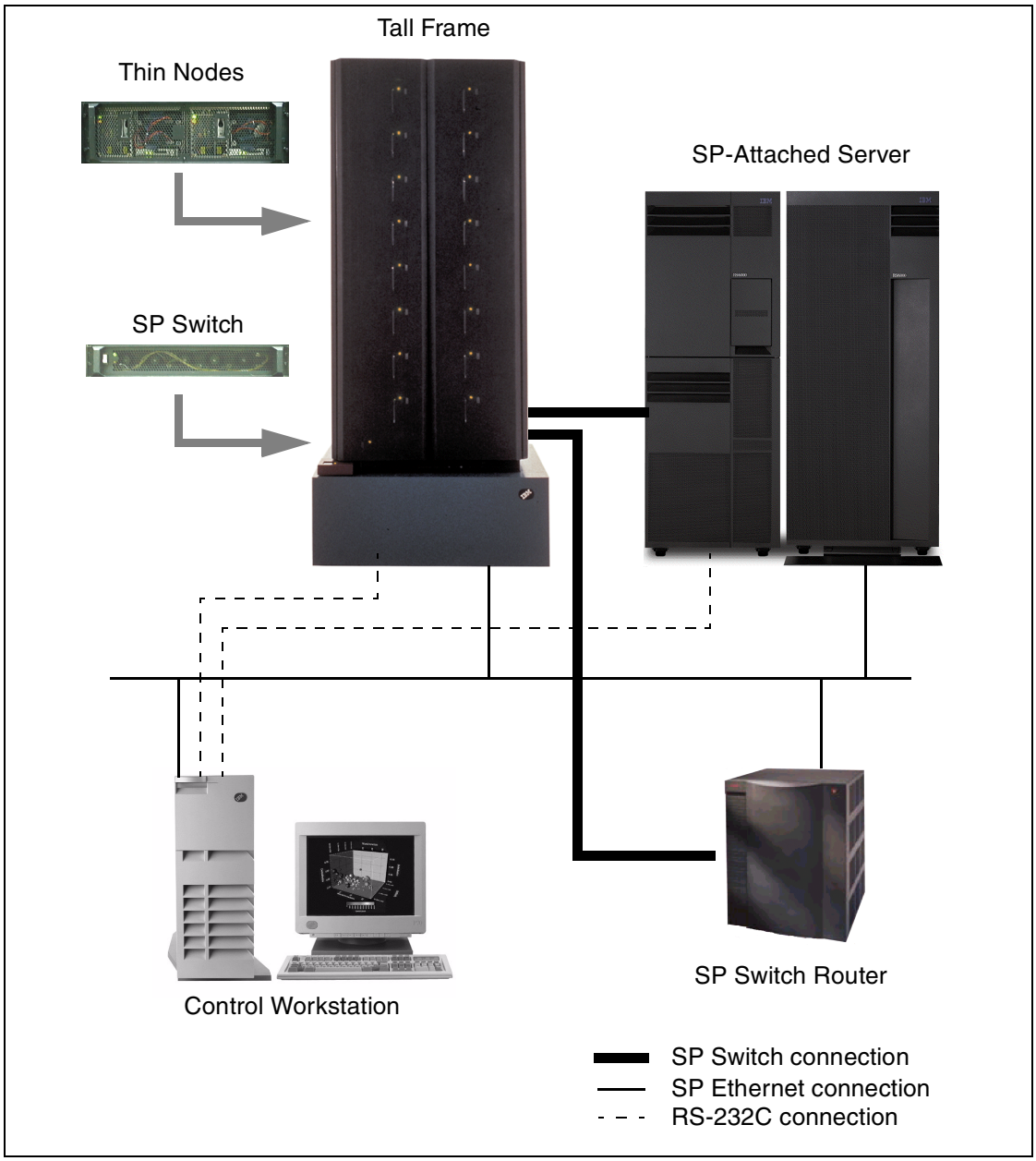


Figure 50. Hardware components in an RS/6000 SP

4.3.3 SP Switch

The SP Switch is, in essence, a high speed communication network. The switch provides a message-passing network that connects all processor nodes with a minimum of four paths between any pair of nodes. The SP Switch can also be used to connect the SP system with optional external devices. There are two RS/6000 SP Switch types:

SP switches
Available as either the 16-port SP Switch or the 8-port SP Switch-8:

- SP Switch2, 16-port switch
- SP Switch, 16-port switch
- SP Switch-8, 8-port switch

High-performance switches
The High Performance Switches are being phased out and are only available for MES upgrades to existing systems. These switches are available as either the 16-port High Performance Switch or the 8-port HiPS LC-8 switch.

4.3.4 Control workstations

The RS/6000 SP system requires a RS/6000 workstation with a monitor. The control workstation serves as a point of control for managing, monitoring, and maintaining the RS/6000 SP frames and individual processor nodes. A system administrator can perform these control tasks by logging into the control workstation from any other workstation on the network.

The control workstation also acts as a boot/install server for other servers in the RS/6000 SP system. In addition, the control workstation can be set up as an authentication server using Kerberos.

4.3.5 Extension nodes

An extension node is a non-standard node that extends the SP system's capabilities but can not be used in the same manner as a standard node.

One type of extension node is a dependent node. A dependent node depends on SP nodes for certain functions, but implements much of the switch-related protocol that standard nodes use on the SP Switch.

A specific type of dependent node is the IBM 9077 SP Switch Router. The 9077 is a licensed version of the Ascend GRF switched IP router that has been enhanced for direct connection to the SP Switch. These optional external devices can be used for high speed network connections or system

scaling using HIPPI backbones or other communications subsystems, such as ATM or 10/100 Ethernet.

4.4 Software components

Software components in an SP are displayed in a layer model, as shown in Figure 51. The layers that are covered are the basic ones from a conceptual and management point of view.

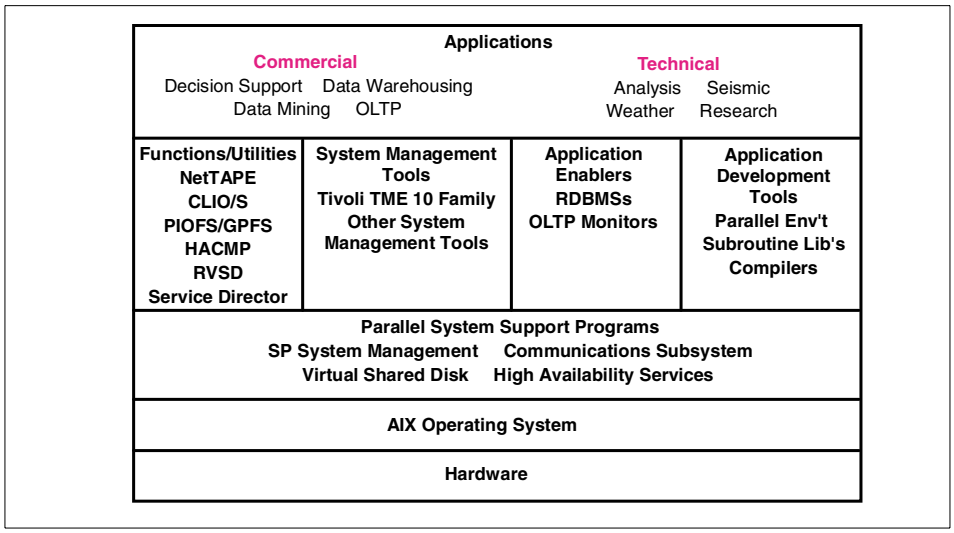


Figure 51. RS/6000 SP system software architecture

Above the hardware layer is the AIX Operating System, which makes the machine's resources available for useful work. The following layer is the PSSP layer, which provides a number of services that help the administrator look at the SP system as one entity. It provides services that help the system be robust, reliable, and available.

Following the PSSP layer is the set of tools and applications running over the PSSP software. In this section, we will briefly describe some of the components of this model in order to give the reader an overview of the SP software architecture.

4.4.1 System Data Repository (SDR)

The SDR is a central repository that contains the specific SP-configuration information. It only resides on the Control Workstation, where the configuration information is then distributed to the nodes.

The SDR stores SP configuration and operational information such as:

- Information about the frames, nodes and switches and how they are configured.
- Job Management operational data.
- Disk configuration information.
- The values of host_responds and switch_responds, two indicators of the status of the node.

4.4.2 Hardware control subsystem (hardmon)

This software controls and monitors some aspects of the SP hardware. There are three hardware components of the SP that can be monitored and controlled:

- Frame - Hardware Control Subsystem monitors electrical and environmental conditions, such as voltage, current temperature, power supply status, and serial link (RS-232) status. SP software enables you to control the power of the frame.
- Node - It monitors the three-digital displays and environmental and electrical conditions, such as temperature, voltage, power, and fan failure. The SP software allows you to control the node's power, key position, and reset button.
- Switch - It monitors environmental and electrical conditions, such as power, voltage, temperature, and fan failure. The SP software allows you to control the multiplexor clock setting and the state of the power of the switch.

4.4.3 Switch software

It is important to understand that although the switch provides data communication across the nodes in much the same way as Ethernet, Token Ring, FDDI, or ATM, it works on an entirely different operating principle.

The SP switch supports the following protocols to communicate between the nodes:

TCP/IP protocol suite TCP/IP is utilized in socket communication for many commercial applications, and is the basis for popular network protocols and Web-based services, such as

Hyper Text Transfer Protocol (HTTP), Network File System (NFS), and File Transfer Protocol (FTP), and applications such as Distributed Computing Environment (DCE). With the possible exception of applications that depend on network-specific functions (for example, LAN broadcasts), most applications work over the switch without modification.

User space protocol The user space protocol is sometimes referred to as a lightweight protocol because it requires fewer processor cycles to transmit a given amount of data, compared to heavier protocols like TCP/IP. User space is most commonly used for scientific and technical computing applications via a message passing interface or the Low-level API (LAPI).

The switch network function is supported by the service software known as the Worm. This service software is coupled with the Route Table Generator, which examines the state of the Switch as determined by the Worm execution, and generates valid routes from every node to any other node. These two components are implemented in the fault-service daemon (fault_service_Worm_RTG_SP), which runs on every node in an SP complex and is central to the functioning of the switch.

4.4.4 Time service

To present a single system image to distributed applications and system management layers, consistent time across nodes is critical. The PSSP partition-sensitive daemons use time stamps in their communications and will quickly become confused if time is not synchronized. Service tickets are keys for Kerberos, the SP authentication mechanism, which tolerates five minutes of difference in the time setting of communicating hosts. Original SP nodes did not have a system battery to preserve the system clock across a shutdown; therefore, a time management protocol was mandatory.

Network Time Protocol (NTP) is public domain code that scales well. The SP can support different applications across different sets of nodes, so a flexible time service is needed; for example, a customer may need to implement two different time zones, corresponding to two applications running on two independent node groups. Different NTP domains can easily be configured within the SP. With the entire SP synchronized, any internal timestamp for an application or management function is the same at any given moment on any node. NTP is optional, and a customer can implement another time protocol.

4.4.4.1 NTP overview

NTP uses a hierarchical, master-slave configuration. In Figure 52, the top of the hierarchy shows the primary servers, which maintain their own clocks. Below the primary servers are the secondary servers. There may be many levels (hops) in the hierarchy of secondary servers. Each level is referred to as a stratum, with the primaries at stratum 0, the first level of secondaries at stratum 1, and so on. The stratum of an NTP server uses a hierarchical, master-slave configuration. The stratum of a server specifies exactly how many hops are required to the primary clock source. The higher a server's stratum number, the more hops are required up the hierarchy, and the less accurate that server's time value.

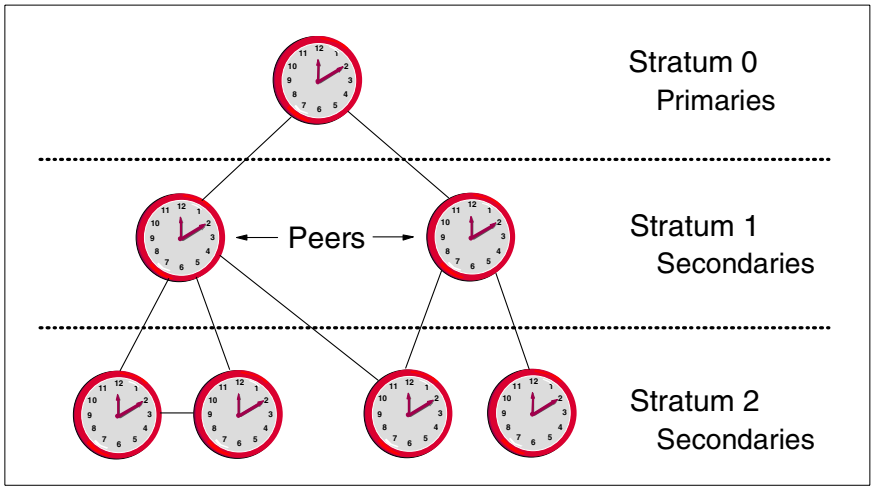


Figure 52. NTP hierarchy

Secondary servers are both clients of lower-stratum servers and servers to higher-stratum servers. Servers in the same stratum are called peers, and may give or request time from each other. A host configured to have multiple time servers employs a selection algorithm to determine the most accurate and reliable server with which to synchronize. The accuracy of synchronization depends on a host's position in the stratum and the duration of synchronization. The duration of synchronization refers to how frequently a host measures to resolve the skew, or clock drift, among clocks. The more frequently the host compares its clock to its time server(s), the better the accuracy.

NTP servers communicate via UDP/IP messages; thus, any IP-capable network can form an NTP hierarchy. NTP servers are IP hosts with corresponding IP addresses. NTP is implemented by the xntpd daemon.

4.4.5 High Availability Infrastructure (HAI)

The High Availability Infrastructure (HAI) provides a set of services to monitor and manage the availability of applications on the SP. HAI is designed to be a cross-platform set of services, potentially implemented on IBM and even other manufacturer's server platforms. The HAI was renamed to RS/6000 Cluster Technology (RSCT) and packaged different from release 3.1 of the PSSP. In this way, non-SP RS/6000 customers could implement HAI.

4.4.5.1 HAI architecture overview

It is out of the scope of this redbook to explain the HAI concepts. There are books entirely dedicated to this purpose:

- SP Monitoring: Keeping it Alive, SG24-4873
- PSSP: Event Management Programming Guide and Reference, SC23-3996
- PSSP: Group Services Programming Guide and Reference, SC28-1675

We deploy some basic concepts about the components of HAI, as shown in Figure 53.

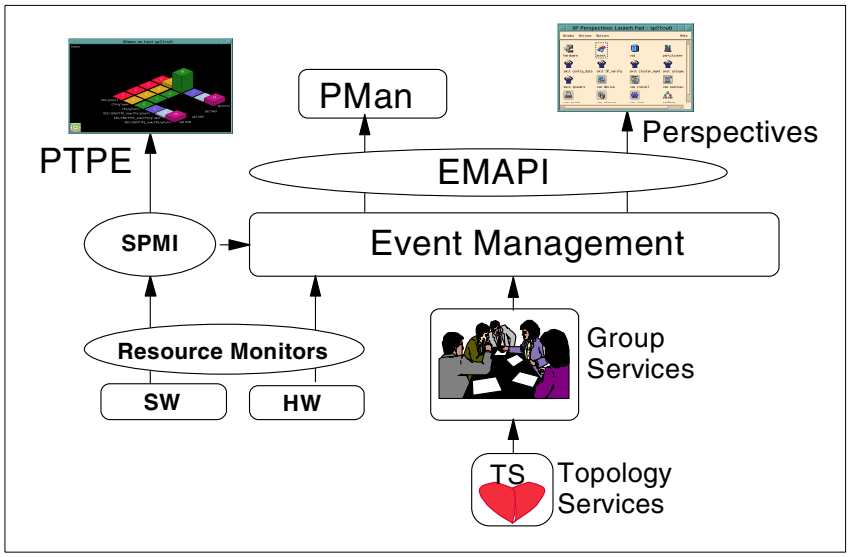


Figure 53. HAI components

Although many components are presented in the figure, HAI comprises three principle elements:

Topology Services (TS) TS maintains availability information about nodes and their network adapters. TS considers a node to be "up" if it can be reached through at least one communication path. Determining valid communication paths is why TS concerns itself with network adapters. Besides, TS maintains the network road map, or Network Connectivity Table (NCT), for use by Group Services' (GS) Reliable Messaging component.

Group Services (GS) GS is a general purpose facility for coordinating and monitoring changes to the state of applications running on a set of nodes. The communication routes selected are determined from the NCT created by TS.

Event Management (EM) EM provides an application with comprehensive monitoring of hardware and software *resources* in the system. A resource is simply an entity in the system that provides a set of services. CPUs execute instructions, disks store data, database subsystems enable applications. You define what system events are of interest to your application, register them with EM, and let EM efficiently monitor the system. Should the event occur, EM will notify your application.

Sitting above EM is the Event Manager Application Programming Interface (EMAPI). Through this API, higher level components gain knowledge of the state and availability of system resources. Examples of these components include Problem Management (PMAN), a system monitor such as the Perspectives graphical user interface, and others.

Feeding EM with SP resource information are Resource Monitors (RM) and the System Performance Measurement Interface (SPMI). SPMI is a construct of Performance Agent. SPMI and EM supply data to higher level performance monitoring tools, such as Performance Toolbox Parallel Extensions (PTPE).

RM, SPMI, and PTPE are SP-specific implementations of resource and performance monitoring. They are not strictly part of base HAI. They change

as HAI moves from platform to platform, whereas TS, GS, and EM are designed to be platform-independent services.

4.4.6 SP security

Security is necessary to protect the SP system from intruders and to keep the SP users from running non-authorized tasks in your system.

There are some security-related concepts that we have to explain to understand the SP security schema:

Identification The process by which an entity tells another who it is by providing the other party with some sort of credentials.

Authentication The process by which the other entity verifies this identity.

Authorization The process performed by an entity to check if an agent, whose identity has previously been authenticated, does or does not have the necessary privileges to carry out some action.

Impersonation To assume another machine's identity or another user identity.

The AIX operating system covers the first three issues, providing the basic security elements, such as access control for user accounts, files and directories, devices, networks and so on.

Impersonation problems are covered in SP systems by authenticating the packages that cross the internal networks. The mechanism used to cope with this issue is Kerberos.

For further information about Kerberos and others elements of security in the SP environment, you can refer to these related redbooks and publications:

- The RS/6000 SP Inside Out, SG24-5374
- Inside the RS/6000 SP SG24-5145
- Exploiting RS/6000 SP Security: Keeping it safe, SG24-5521

4.4.7 Automounter

The automounter is used for automatic and transparent mounting and unmounting of NFS file systems. When a user accesses a file or directory under automounter control, automounter automatically mounts the file system and, after some determined amount of time, the file system is unmounted.

There have been two different versions of automounter supported by PSSP software: the BSD Automounter and AIX Automounter. The BSD Automounter, also known as AMD, was replaced by the AIX Automounter in PSSP Version 2.3, because it presented many problems in the field.

The AIX automounter is a version of the SunOS 4.x Automounter. Its daemon software is part of NFS in the Network Support Facilities of the AIX Base Operating System (BOS) Runtime, and is fully supported by AIX.

4.4.8 Parallel I/O

Applications can exceed the I/O capacity of a single node. In other words, applications may need access to data that is spread across multiple nodes in order to improve the I/O performance of the system. Any node should be able to access data residing on any other node. The SP can solve this problem with these different solutions:

- Network file systems (NFS)
- Andrew file systems (AFS and distributed file systems (DFS))
- Virtual shared disks (VSD)
- Hashed shared disk (HSD)
- Recoverable virtual shared disk (RVSD)
- General parallel file system (GPFS)

4.4.8.1 Network file system

Network File System (NFS) is widely used on the SP. It is a component of AIX and is exploited by the node installation and customizing process. It is very flexible, easy to install and configure, and can be more available with products as High-Available NFS (HANFS).

The NFS's problems are restrictions with scalability, because an NFS server only can reside in a single node, and with the performance.

4.4.8.2 Andrew file system and distributed file system

Andrew File System (AFS), from Transarc, and Distributed File System (DFS), from the Open Software Foundation (OFS), offer more secure, manageable file system implementations, compared to NFS. They offer little more in the way of I/O scalability and performance, as they fundamentally remain a mechanism for distributed data access.

4.4.8.3 Virtual shared disk

Virtual Shared Disk (VSD) allows data in logical volumes on disks physically connected to one node to be transparently accessed by other nodes. Importantly, VSD supports only raw logical volumes, not file systems.

4.4.8.4 Hashed shared disk

While VSD enables efficient distributed access to raw logical volumes, the performance of any VSD is restricted by the I/O capacity of the VSD server node. To improve I/O bandwidth, IBM provides Hashed Shared Disk (HSD) in the base PSSP. HSD is a short version of VSD. If the I/O load on a specific VSD is too heavy, you can use HSD to distribute the load across other VSDs and nodes.

4.4.8.5 Recoverable virtual shared disk

Recoverable Virtual Shared Disk (RVSD) adds availability to VSD. RVSD allows you to twin-tail disks, that is, physically connect the same group of disks to two or more nodes, and provide transparent failover of VSDs among the nodes.

4.4.8.6 General parallel file system

VSD, HSD and RVSD improve the flexibility, performance, and availability to access to large amounts of data on the SP, but only for raw logical volumes. General Parallel File System (GPFS) addresses all requirements for file systems.

GPFS provides a standard, robust file system for serial and parallel applications on the SP. From a user's view, it resembles NFS, but unlike NFS, the GPFS file system can span multiple disks on *multiple nodes*. GPFS exploits VSD technology and the Kerberos-based security features of the SP, and is only supported on SP systems.

4.4.9 File collection

File Collection is used to simplify the task of maintaining duplicate files on multiple nodes of the SP system. In a standard SP system, the files that are required on the Control Workstation, boot/install servers, and processor nodes, belong to file collections. By grouping these files into file collections and using the provided tools, you can easily maintain their consistency and accuracy across the SP nodes.

The Parallel System Support Program (PSSP) is shipped with a program called supper. The supper has a set of subcommands. The files in a collection are handled with special procedures using these `supper` commands. The

`supper` commands interpret the master files and use the information to install or update the actual files in a collection.

A file collection can be either Resident or Available. A Resident file collection is one that is installed in its true location and able to be served by other systems. A file collection that is not installed in its true location but is able to be served by other systems is called an Available collection.

File collections can be either primary or secondary. Primary file collections can contain secondary file collections. When a primary collection that contains secondary collections is installed, the secondary collections are available but not resident.

4.4.10 Job management

In a system where you have many nodes in a machine, you may need to have a mechanism to share batch work on the various nodes. If not, you can have some nodes not used at all, and other nodes overloaded.

SP provides several tools to manage this issue: the AIX job management and LoadLeveler.

4.4.10.1 AIX job management

All the traditional AIX commands that are used to handle job management are still available when AIX is running on a node. Here is a list of AIX commands for job management.

- `at` allows the system administrator to specify a shell script to be run once at a specific time. In this shell script, one specifies which program or programs to run and on which node or nodes. This script has to be well written to handle all possible error conditions.
- `batch` could be used in conjunction with `at` to specify a list of jobs that will be run only when the processor load level allows it.
- `cron` is used on each node to specify a shell script to be run at a regular time. This command, like the others, uses shell scripts that have to be well written to handle all possible error conditions.

These commands alone are not sufficient to effectively manage more than one node.

4.4.10.2 LoadLeveler

LoadLeveler allows one to conceptually submit all the jobs that need to be run into a pool, and then distributed them out to the various nodes in an efficient manner.

First, specify which node is the primary Central Manager, and then nominate the alternate Central Manager to ensure availability of the system. Usually, the control workstation is the primary Central Manager. The Central Manager collects the various jobs and delegates them out to specific nodes.

Categorize the various nodes by the type of work that they are best suited to perform, and categorize the jobs by the types of nodes that they run on. One can also specify nodes to be submit-only nodes, so these nodes do not receive any work from the Central Manager.

Then categorize the various users and groups according to:

- Their job priority.
- The limit on the number of jobs that they can submit at a time.
- The type of job to be run as a default if none is given.

Once the nodes that do the work and the users that submit the work have been classified, it only remains to classify the kinds of work that can be submitted. A whole list of commands have been developed to manage and use LoadLeveler, but there is also an easy-to-use Graphical User Interface (GUI). This GUI assigns different colors for the activity levels of all the nodes.

4.4.10.3 Resource manager

The problem with LoadLeveler is that it only directly handles serial jobs, or jobs that run on one node at a time. When the job request indicates that the job must be run in parallel, LoadLeveler sends the request to the Resource Manager, who takes care of delegating the job to the required nodes. See Figure 54 for more details.

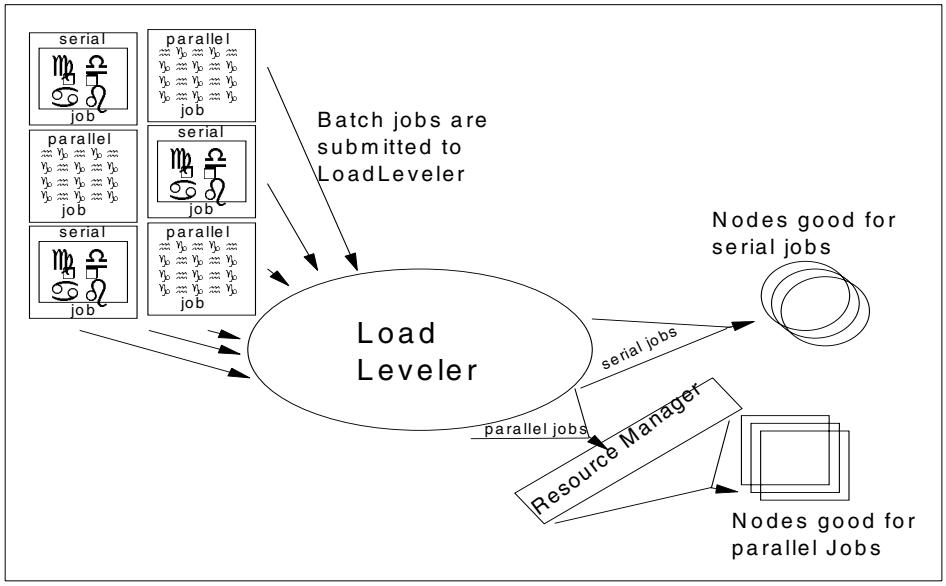


Figure 54. Interaction of LoadLeveler and Resource Manager

The Resource Manager is a program that runs on one of the nodes. It prevents parallel jobs from interfering with each other and reports job-related node information back to LoadLeveler. The system administrator has to add a couple of options to the configuration of LoadLeveler to enable it to interface with Resource Manager.

4.4.11 Parallel environment

IBM Parallel Environment presents a UNIX-like serial program interface to the end user, while providing powerful message passing libraries optimized for the RS/6000 SP. It also provides tools for compiling, debugging, and analyzing the performance of parallel applications. Parallel Environment supports both interactive and batch execution via LoadLeveler.

Parallel Environment includes components and tools for developing, executing, debugging and profiling parallel programs. These are:

- Parallel Operating Environment (POE)
- MPI/MPL Message Passing Libraries
- LAPI (Low_level API)
- Parallel Debuggers
- Xprofiler Profiler Analyzer

4.5 Benefits of using the SP in the ISP arena

At this point we have briefly described the SP characteristics, and we find advantages if we implement our ISP inside an SP.

The following SP's features could help us in the ISP installation, deployment and management:

- Manageability
- Scalability
- Flexibility
- Availability
- Performance

4.5.1 Manageability

The single point of control of an SP system is the CWS. From the point of view of monitoring, you have a graphic panel where you can control the current status of all your servers.

Software updates can be done in a more effective and faster way using the CWS as a installation server and the internal network to update software concurrently.

New product installations or node reinstallation and manageability issues, such as microcodes updates, and powering on and powering off the nodes are easier and faster using the parallel commands provided by PSSP environment.

From the point of view of maintenance, SP system has hot pluggable capacities in two components, nodes and frames. You can remove, add, or replace some components without stopping the service, if the component is not vital for the system, such as the frame's power (which is a redundant component), network cards, and SSA disks.

All the described advantages and how your system can implement them will be shown in Chapter 6., "Sample implementation" on page 219.

4.5.2 Scalability

The scalability in SP systems is part of their design philosophy. It is a process well documented and tested with customers and laboratories around the world. If you need to add more nodes either to increase the performance or your ISP services, or you need to add a new development environment, you

can do it without much difficulty. The main reason is due to the fact that the CWS can manage the new infrastructure just as your initial configuration.

The scalability is bidirectional: up and down. You can add new services or increase the performance of your ISP by adding new nodes or frames, and you can decrease the performance or eliminate some services by removing nodes or frames and assigning them to other purposes, using the manageability features of this system.

4.5.3 Flexibility

At the moment, we have spoke about scalability in both directions and manageability. These features provide us with a system that has a high level of flexibility when we want to add more services to our SP or split services between different nodes.

These problems are solved in an effective way by using parallel commands provided by PSSP software and the particular configuration that let us control the hole system from a single point of control.

4.5.4 Availability

In this business, a 24X7 availability is required in the servers if we want to provide a high satisfaction to the customers.

The SP system is fully compatible with all the techniques, products and tools available for the AIX operating system that can help us maintain and run the system when a hardware failure occurs. Tools are widely known in the marketplace by their reliability and performance in high availability environments. We can improve our system availability with products and technologies, such as HACMP, Edge Server Load Balancer (ELDB), disk mirroring, or SSA disk technologies.

A single point of failure on the SP system could be the CWS, the only point of control of the SP system, but there are special products designed to provide high availability to this component based on HACMP technologies (HACWS). This component provides a specific HACMP solution to ensure the high availability of the CWS.

4.5.5 Performance

The performance in an ISP is related to the response time to user requests. Good responses times are important for user satisfaction and are related to the application design and the performance of the software and hardware involved in the solution.

SP system performance could be improve in several ways. Increasing node performance, adding processors or memory (if they are overloaded). The node may be at the limits of its computing capacity; in this case, you can add an attached S80 server, which is the most powerful server in the marketplace.

You can use the switch bandwidth to interconnect the nodes, or distribute the load between the nodes, in order to complete complex tasks when using Oracle Parallel Edition or DB2 Parallel Server. The switch can be connected to a switch router (GRF) to provide high bandwidth access to the SP internal switch network (if the solution design requires this capability), therefore simplifying the network design and avoiding additional network cards inside nodes.

Chapter 5. How do ISP components fit in an RS/6000 SP?

ISPs have different requirements, so it is difficult to define a generic structure for the ISP platform. However, there are some basic characteristics that all ISPs have. We have identified the basic requirements for an ISP in Section 1.3, “Requirements for ISPs” on page 15 and then presented the overall architecture of ISPs in Chapter 2, “Overall architecture for Internet Service Providers” on page 27. We will now describe three generic examples for ISP solutions on the SP. We based our scenarios on the segmentation characteristics defined on Section 1.2.4, “Segmentation” on page 4. The three basic models are:

- Market visibility model
- Basic Internet access with basic services model
- Managed e-business services model

In this chapter, we try to describe these models on a physical level and show how we can fit them inside an RS/6000 SP system.

5.1 Architectural definitions and representation of the models

The models presented in this chapter are not complete operational architecture definitions. However, for each basic model and their enhanced model, we indicate the set of functionalities and capabilities that they can provide.

The following identifies the three architectural models, and their enhanced characteristics:

- Market visibility model

  This model is designed for a quick startup with minimal services.

  - Enhanced market visibility model

    The enhanced market visibility model provides some authentication capabilities and user customization.

- Managed Internet access with basic services model

  The services provided in this model are oriented to subscribers that have enrolled in the ISP.

  - Enhanced Internet access with basic services model

The Enhanced Internet access with basic services model provides more services and connectivity capabilities. This model provides the support to WAP (Wireless Application Protocol) devices.

- Managed e-business services model

  The managed e-business services model is based on the front-end/back-end paradigm. This model provides platform and security features for difficult business requirements.

  - Enhanced e-business services model

    The Enhanced e-business services model adds connectivity capabilities and services. This model provides the support to different pervasive devices.

All the models are prototypes, not real scenarios, but all of them have common characteristics with real solutions. The Market visibility model is a simple solution that is more or less a starting point for the ISP platform development. We provide installation guidelines for the enhanced market visibility model in Chapter 6, “Sample implementation” on page 219. The basic Internet access with basic services model provides the Internet services for residential customers. This model is the entry point to the real ISP implementations. The managed e-business services model provides a good base for ISP and ASP (application service provider) implementations. This environment is more complex than the first two models, but it also provides answers to difficult business requirements.

5.1.1 Considerations about the hardware configuration selected

All the configurations proposed use thin nodes to simplify the functionality description and the growing path selected in each case. Other configuration possibilities are:

- Use more powerful nodes

  Wide and high nodes improve the computing capability of thin nodes, and provide expansion possibilities. It is possible to group several functions inside a high node instead of using four thin nodes. However, in order to clarify the functions included in the ISP, we divided the functions between several thin nodes.

- Use SP-attached servers

  The most powerful server in the marketplace, the S80, is attachable to an SP frame and installed, managed, and administered by the CWS as any other node in the SP. If the ISP requires high computing power for certain services, you can consider attaching an S80 to the RS/6000 SP.

- Use the SP Switch internal network

  Under certain circumstances, using the SP Switch, an internal high bandwidth network to exploit the parallel capabilities of the SP system, may be useful. There are solutions that may require this high performance network for some functions, for example, parallel database management systems with high degree of complexity or systems backup.

5.1.2 Considerations in the architecture

In the architectural models, we have avoided direct reference to the ISP products, and included only the functionalities required for providing ISP services.

In some situations, there are products that offer several different functionalities for the ISP. In these cases, the product is introduced and described.

The load balancing function is a special issue in this book. The IBM product that provides this function is the Edge Server Load Balancer (ESLB), referred to in Section 3.2.5.1, “Edge Server Load Balancer (ESLB)” on page 118, and previously known as eNetwork Dispatcher (eND). We will refer to the eND product throughout this publication, because its functionality is widely known in the ISP community.

5.1.3 Access network assumptions

In the models presented in this chapter, the a*ccess network* will not be covered in the architecture description, logical description, or in the physical description. The access network is described on Section 2.1.2, “Access network” on page 38. We made the assumption that the ISPs already have the basic network infrastructure for the dial-in connection and the Internet connection. The Internet connection is essential for the ISP to achieve market visibility and launch their services. The dial-up connection service must be in place before the ISP starts providing services.

5.2 Market visibility model

The characteristics of this model are to provide quick market visibility in the Internet, and to provide basic information about the activities of the business. This model requires the minimum implementation components for an ISP.

5.2.1 Logical architecture

The market visibility model provides the minimum service for an ISP. Basically, it provides the channels for marketing and advertising, but not billable services for the customer. Free mail service is one way to raise users interest in the ISP. In order to provide the initial channels for marketing and advertising, we will need to implement some components to ensure availability, load balancing, and security. The logical architecture and the components for the market visibility model are shown in Figure 55.

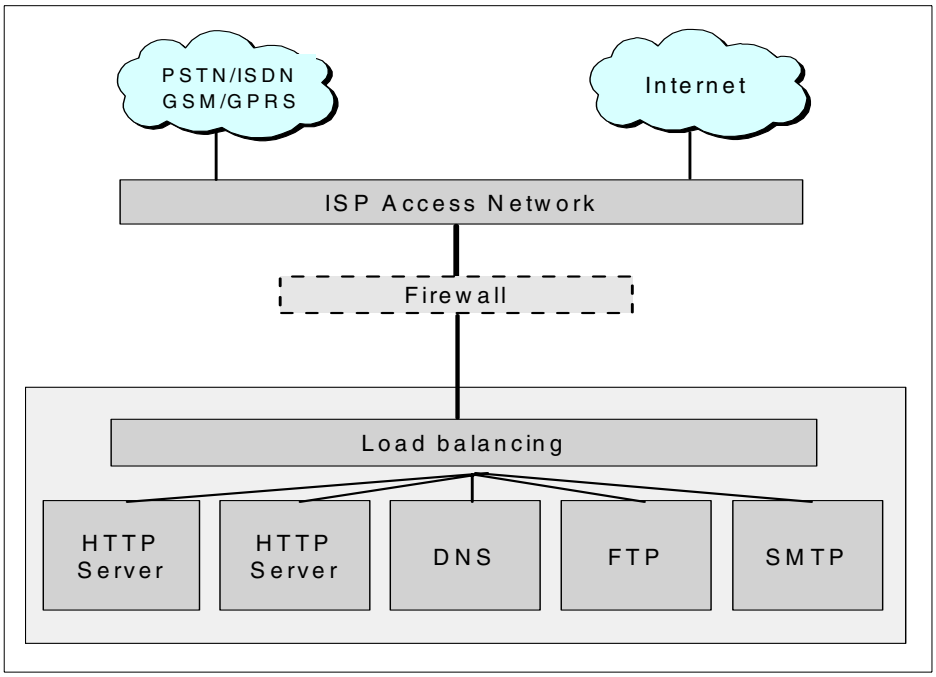


Figure 55. Logical architecture for the market visibility model

The logical components for the Market visibility model are:

Access network It is not described in our architecture, because is not an issue in the SP system.

Firewall Security component to avoid unauthorized access.

Load balancing Improves performance of the ISP by balancing the client requests between the HTTP servers, and ensures availability if one of the servers is down.

HTTP servers Provide static Web pages, mostly for marketing, to the clients connected to the ISP.

FTP server The File Transfer Protocol (FTP) service provides the users the capability to download files from the ISP.

Mail server The mail server provides mail accounts to the subscribers of the ISP via the SMTP service protocol.

DNS services The Domain Name Server (DNS) makes the ISP visible in the Internet.

5.2.2 Physical architecture

From the physical point of view, our *Market visibility model* is a single frame with four nodes and one Control Workstation (CWS), as shown in Figure 56. The model includes a basic security configuration to protect the DMZ zone, and one point of control based on the CWS.

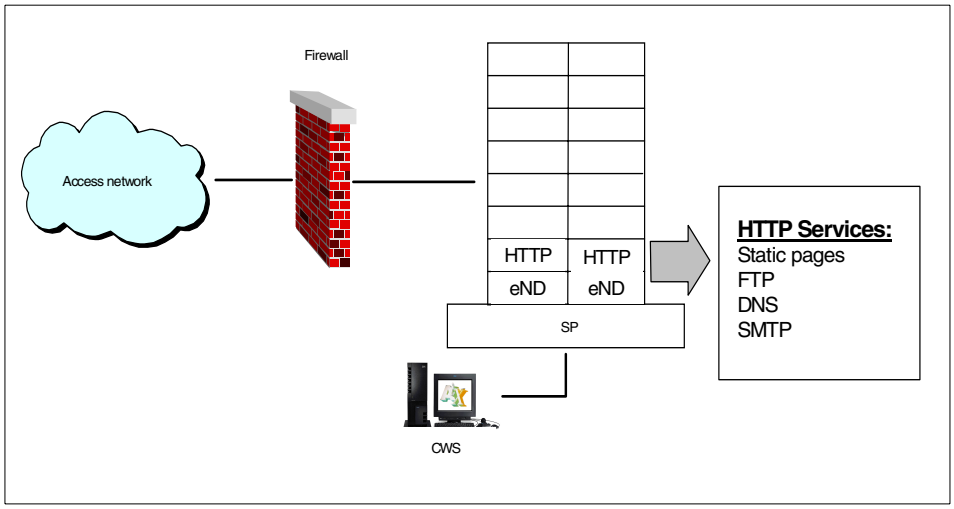


Figure 56. Physical architecture for the Market visibility model

The services are distributed between four nodes to ensure load balancing and high availability. Two nodes are dedicated to load balance the users' requests (eND nodes) in a high availability configuration. The DNS, FTP, and SMTP services will be provided by the HTTP server nodes.

From the point of view of security, we need only one firewall to protect the servers in the DMZ from external attacks. This implementation is oriented to get a fast and scalable solution to have visibility in Internet.

5.2.3 Limitations

This model has limitations derived from its small size and simplicity:

- It provides a reduced number of services.
- Performance may suffer if the number of users increase.
- No possibility to store data or create dynamic Web pages.
- No possibility for customer care functions.
- No connection is possible with other legacy applications.

We will try, with the next models, to solve some of the limitations found in the market visibility model.

5.3 Enhanced market visibility model

The Market visibility model can grow in several possible directions:

- Manage a higher number of hits from the Internet by improving its performance.
- Adding more services to the ISP.
- Improving connectivity to the ISP.

In order to grow the previous model, we propose the Enhanced Market Visibility Model. The enhanced market visibility model provides added features, such as basic authentication services, dynamic Web page generation capabilities, and high availability capabilities for the new components.

5.3.1 Logical architecture

The logical architecture for the enhanced market visibility model is shown in Figure 57 on page 203.

.

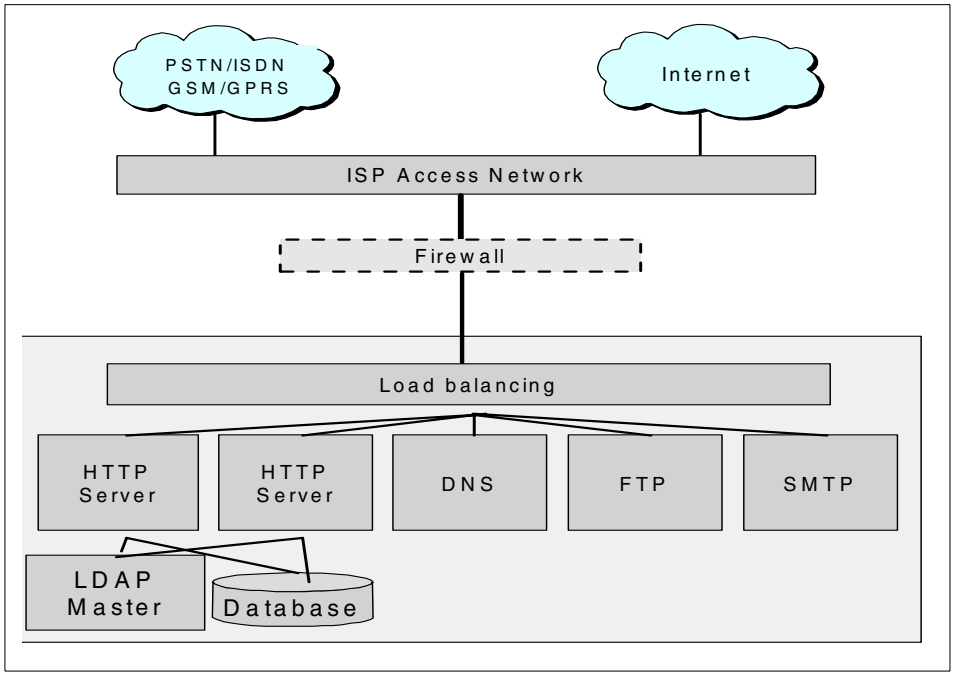


Figure 57. Logical architecture for the enhanced market visibility model

This model provide the following added services:

Authentication Basic authentication services based, on the Lightweight Directory Access Protocol (LDAP).

CGI scripts Common Gateway Interface (CGI) scripts provide some capabilities to generate dynamic Web pages and connecting the HTTP server nodes to the database. This functionality is included in the HTTP servers.

The HTTP servers request server basic authentication services from the LDAP, which are stored in the database.

Security is based on one firewall that separates the DMZ zone from the Access Network.

5.3.2 Physical architecture

This proposal includes two more nodes to implement the authentication service and the high availability components, as shown in Figure 58. We installed a DB engine besides the LDAP software on the two nodes (master and slave), and using the LDAP replication feature, achieve high availability of this service.

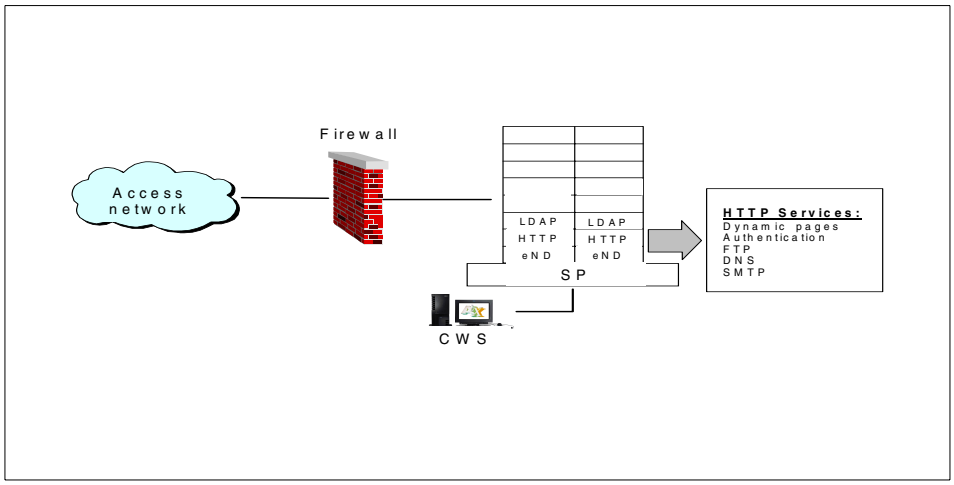


Figure 58. Physical architecture for the enhanced market visibility model

Security has the same schema as in the previous model: A firewall between the access network and our DMZ zone prevents unauthorized access.

5.4 Managed Internet access with basic services model

Managed Internet access with basic services model is the basic ISP structure for Business to Customer (B2C) solution. The model provides some basic ISP services, as explained in Section 5.4.1, “Logical architecture” on page 204.

5.4.1 Logical architecture

The managed Internet access with basic services model provides the following services:

Load balancing Improves performance of the ISP by balancing the client requests between the HTTP servers, and ensures availability if one of the servers is down.

HTTP servers Provides Web pages to the clients connected to the ISP. In this model, we can customize the pages served to the

different users using the authentication capabilities of the LDAP protocol.

FTP server The FTP server provides the subscribers with the capability to download files from the ISP file repository.

Mail server The mail server provides mail accounts to the subscribers under several protocols, such as POP3, SMTP, or IMAP4.

DNS services Domain Name Server (DNS) converts the names provided to users into IP addresses in order to access the services provided by the ISP.

Enrollment Service to automatically enroll new subscribers to the ISP. The enrollment process is made by the subscribers.

RADIUS An authentication server for dial in authentication based on RADIUS. This server confirms subscriber log-ins and passwords.

Web hosting Web hosting provides capabilities to create, maintain and storage subscribers' personal Web pages.

Portal pages Provide personalized portal pages to the subscriber with interesting information for that specific subscriber.

Figure 59 on page 206 shows how the services are distributed in this model. The ISP access network request passes through the firewall, which filters the traffic allowed to request information. This traffic is dispatched, depending on the service required, to different logical servers which consult the subscriber management systems or mail services for what information the user has authorization to access.

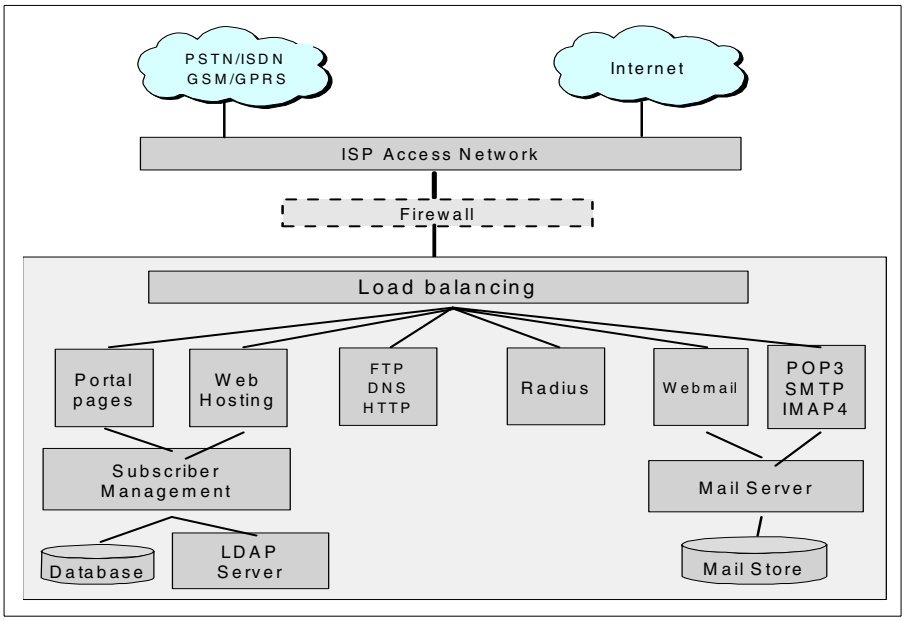


Figure 59. Managed Internet access with basic services model architecture

There are two databases shown, one for the subscriber management and the other for mail server storage. The LDAP server maintains its own database and it is not shown for simplicity reasons.

Security is managed by a firewall in front of the DMZ, isolating the ISP from unauthorized user access.

5.4.2 Physical architecture

This model's implementation is composed of one SP frame controlled by a single CWS, as shown in Figure 60 on page 207.

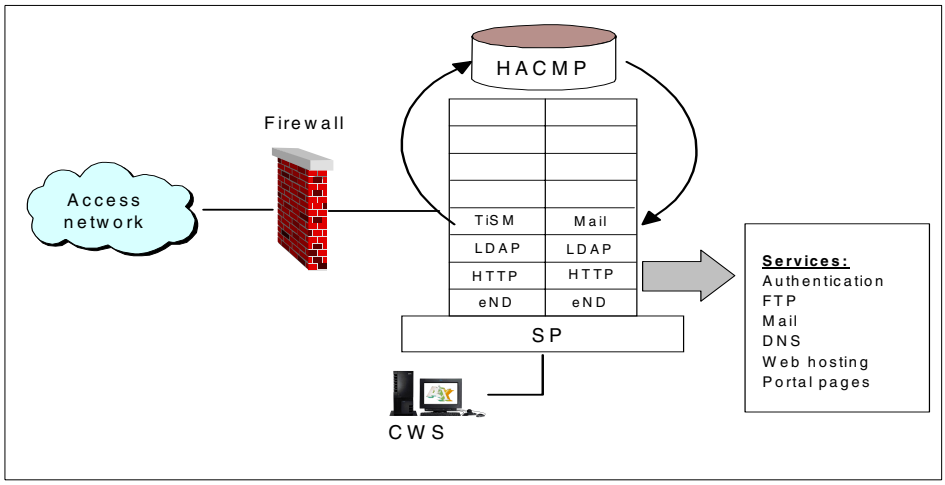


Figure 60. Managed Internet access with basic services model architecture

In this physical architecture, we use the Tivoli Internet Subscriber Manager (TiSM) product to reflect the distribution of our functionalities because it provides an array of services to our ISP.

The services provided by this product in this implementation are:

- Portal pages
- Web content hosting
- RADIUS authentication server
- Subscriber management

More information about this product is available in Section 3.2.8, “Tivoli Personalized Services Manager (TPSM)” on page 125.

One node is running the TiSM and another is running mail. Both nodes are in mutual high availability, to avoid loss of service in case of hardware failure. If the TiSM node is stopped, the mail node will run the TiSM application and vice versa. This configuration provides a degraded performance in case of failures, but there is not interruption of service.

The HTTP services are distributed between two nodes to ensure load balancing and high availability. These nodes also include the DNS and the FTP services.

Two more nodes are required to implement the authentication service with high availability. We installed a DB engine besides the LDAP software on the

two nodes (master and slave), and using the LDAP replication feature, achieve high availability of this service.

From the point of view of security, we only need one firewall to protect the servers in the DMZ from external attacks.

5.5 Enhanced managed Internet access with basic services model

The enhanced managed Internet access with basic services model improves the functionalities of the previous models by adding more services to the ISP. No more frames are required to install the new components in the SP. We only will need to add a couple of nodes to accomplish the task.

5.5.1 Logical architecture

The logical architecture for the enhanced managed Internet access with basic services adds some new services to the ISPs models presented in previous sections. The logical architecture for the new model is shown in Figure 61 on page 209.

We added a component outside the SP architecture to provide the WAP connections to the customers.

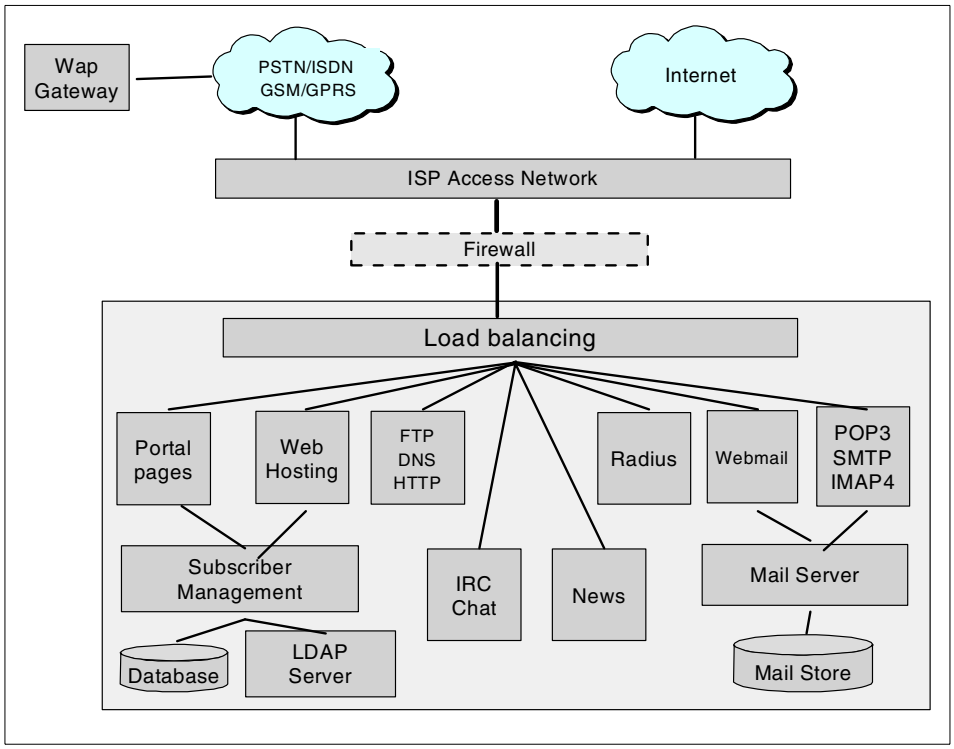


Figure 61. Enhanced Internet access with basic services model architecture

The enhanced managed Internet access with basic services model provides the following additional services:

IRC services Internet Relay Chat (IRC) provides a mechanism to communicate in real time with people from all over the world.

News groups InterNet News (INN) is a set of protocols for exchanging messages in a decentralized network of news servers. Each individual news server stores locally all articles it has received for a given news group.

WAP services Basic WAP services, provided by the WAP Gateway outside the DMZ, routes the requests to the ISP.

Security is managed, as in the previous model, by a firewall in front of the DMZ, isolating the ISP from unauthorized access.

5.5.2 Physical architecture

In this model, we added two more nodes to provide more ISP functions and to enhance the value added services provided by the SP, as shown in Figure 62.

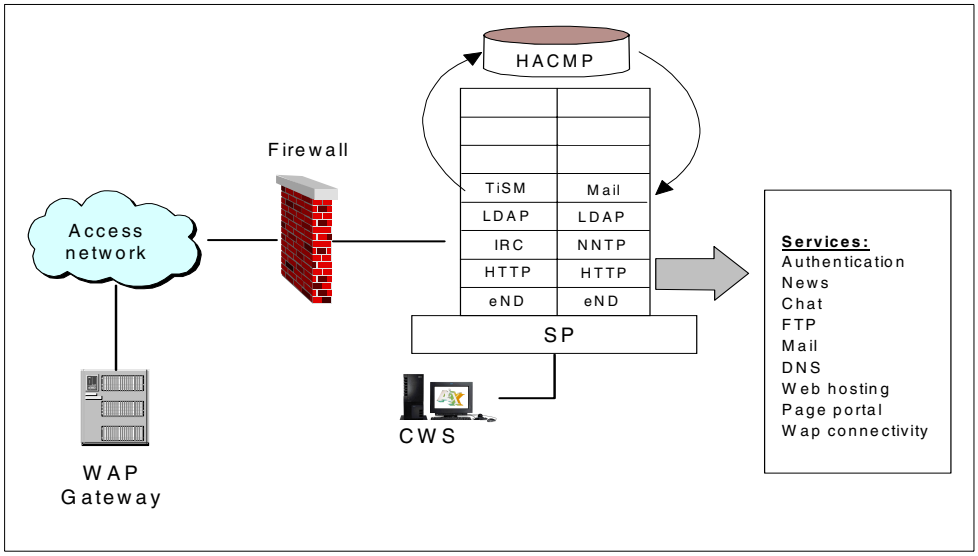


Figure 62. Enhanced Internet access with basic services model architecture

There is one more node to provide news services with the NNTP protocol and another node dedicated to provide IRC services. These services are not configured for high availability, because they are not critical to the operations of the ISP. If the ISP loses these services, the other core services will not be affected.

We need to configure a WAP Gateway outside the SP configuration to provide some basic WAP services.

Security is configured the same as in the managed Internet access with basic services model configuration.

5.6 Managed e-business services model

This model is an approximation to B2B (business to business) with two secure zones defined: the DMZ and the Trusted Network, which connect the core applications and the confidential data of the ISP. The trusted network could be interconnected with other ISPs to build business to business transactions.

5.6.1 Logical architecture

This logical architecture has two different secure zones: the DMZ and the Trusted Network, as shown in Figure 63.

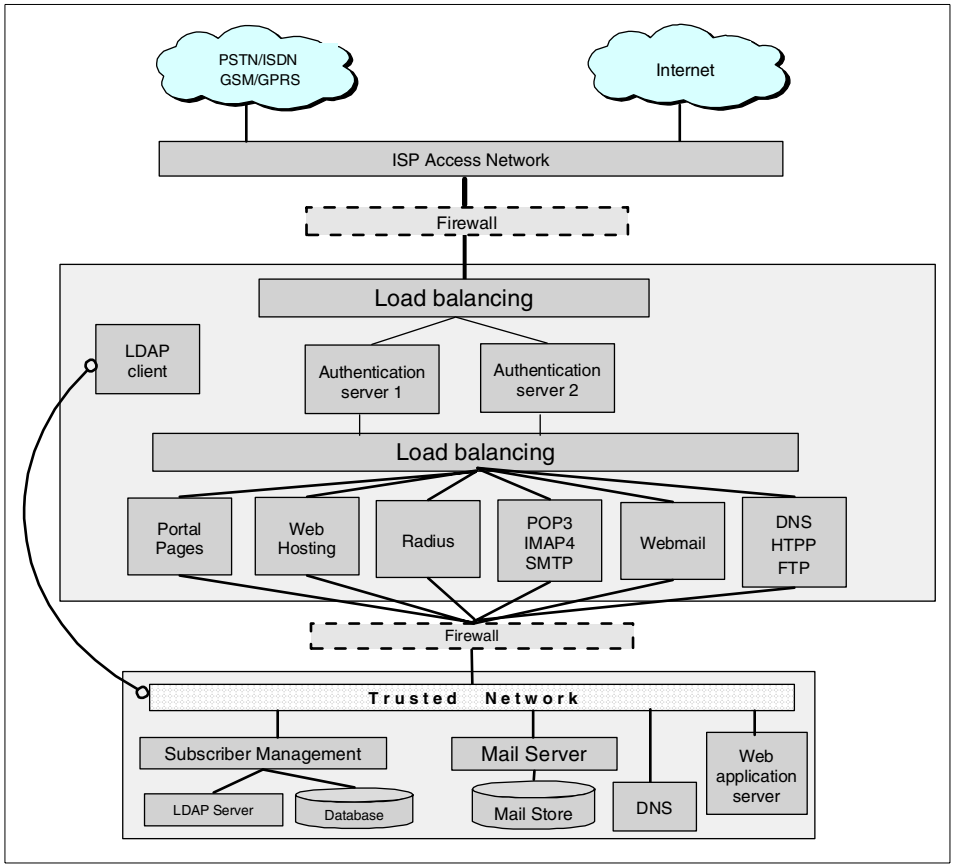


Figure 63. Logical architecture for the managed e-business services model

The following services are located in the DMZ zone:

Load balancing Improves performance of the ISP by balancing the client requests between the HTTP servers and ensures availability if one of the servers is down. Note that load balancing is used in two different levels. The load balancing service connected to the authentication servers provides the authenticated users the services they require.

Portal pages This functionality provides personalized HTML pages to the ISP's subscribers.

RADIUS An authentication server based on the Remote Access Dial In User Service (RADIUS). This server confirms subscriber log-ins and passwords.

Authentication Two authentication servers provide load balancing, and high availability is configured to ensure availability of the service in the event one of the servers is down. The authentication services provide single sign-on for the ISP services. In the DMZ zone, they are located in the front-end of the service, and for the Trusted Network, they are located in the back-end.

FTP server File Transfer Protocol (FTP) server provides the subscribers the capability to download files from the ISP data repository.

Mail server The mail server provides mail accounts to the subscribers under several protocols, such as POP3, SMTP, or IMAP4. It also provides the front-end to the Webmail services.

DNS services Domain Name Server (DNS) converts the names provided to users into IP addresses to access the services provided by the ISP in the DMZ.

Web hosting Web hosting provides capabilities to create, maintain, and store subscribers' personal Web pages.

LDAP client We need an LDAP client connected to the LDAP server to have the same user information in both zones: the DMZ and the Trusted Network.

The following services are located in the Trusted Network:

LDAP server It ensures consistency of the information presented to the ISP users.

Mail server The mail server provides mail storage to all the front-end protocols used in the ISP. It has its own database.

DNS services Domain Name Server (DNS) converts the names provided to users into IP addresses to access the services provided by the ISP in the Trusted Network.

Web Application Server The Web Application Server is the integration service component. The ISP platform can communicate through this component with other systems: content providers, clients, databases, ERP, and legacy applications.

Subscriber Management This service provides authentication service. Web hosting services and portal page services. Subscriber management serves, stores, audits, and manages the information.

Security is managed by two firewalls, one of them protecting the DMZ from the external world and the other protecting the DMZ from the Trusted Network.

5.6.2 Physical architecture

In this model we have done important changes. We have two different zones with two different frames to provide the services to this scenario.

This solution contains two points of control with two different CWS to administrate the independently defined zones. This solution is depicted in Figure 64 on page 214.

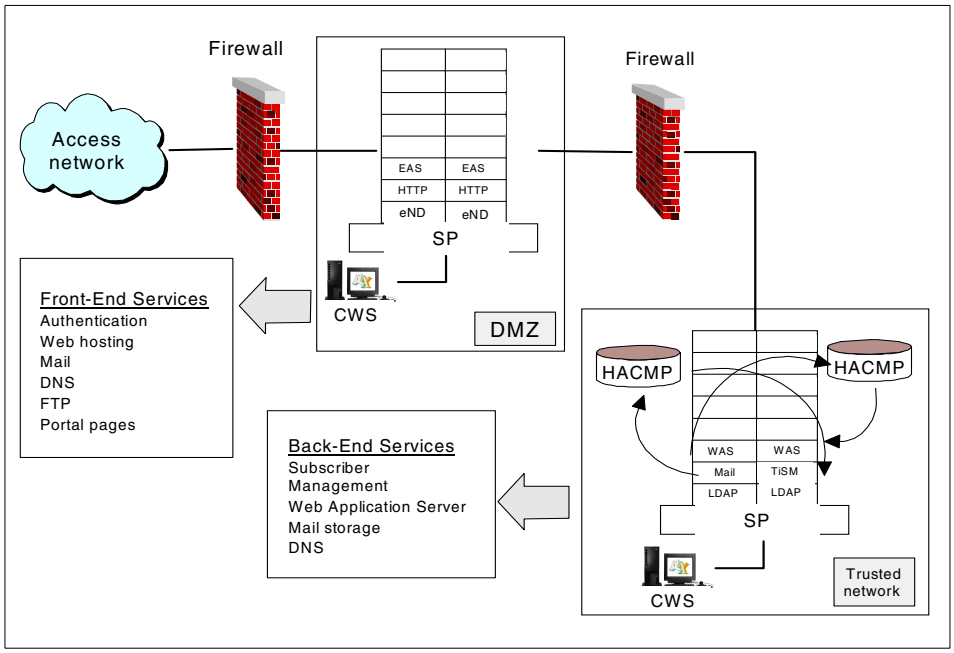


Figure 64. Physical architecture for the managed e-business services model

DMZ frame

Node distribution in this frame is configured as follows:

- Two nodes are dedicated to host the authentication services. This functionality is referred to in Section 3.2.7, “Everyplace Authentication Server (EAS)” on page 124.
- Two nodes host the HTTP servers and provide the HTTP, DNS, FTP, and LDAP client services. They also provide the front-end of the Web hosting, portal pages, mail, and RADIUS services.

Authentication services and HTTP servers are load balanced through the eND nodes to ensure high availability for these services.

Secure Network frame

Node distribution in this frame is configured as follows:

- One node runs TiSM and the other runs the Mail feature. These nodes are in mutual high availability to avoid service interruption in the case of hardware failure. If the TiSM node stops, the Mail node will run the TiSM application and vice versa. This take over configuration provides a

degraded performance in case of failures, but there is no interruption of the service.

- Two nodes are required to implement the LDAP service with high availability. In the two nodes (master and slave), we have installed a DB engine besides the LDAP software and using LDAP replication feature, we achieve high availability.
- The Web application server is hosted in two nodes. The nodes are configured with HACMP and with external disks to achieve high availability.

There are two firewalls to separate the DMZ frame and the Trusted Network frame, ensuring information security in the back end.

5.7 Enhanced e-business services model

The enhanced e-business services model is an improvement on the managed e-business services model because more services have been added to the ISP configuration.

5.7.1 Logical architecture

As shown in Figure 65 on page 216, the logical architecture of this model is not much different than the managed e-business services model. We added some components to provide pervasive device support for the subscribers and additional Internet Relay Chat (IRC) and news services.

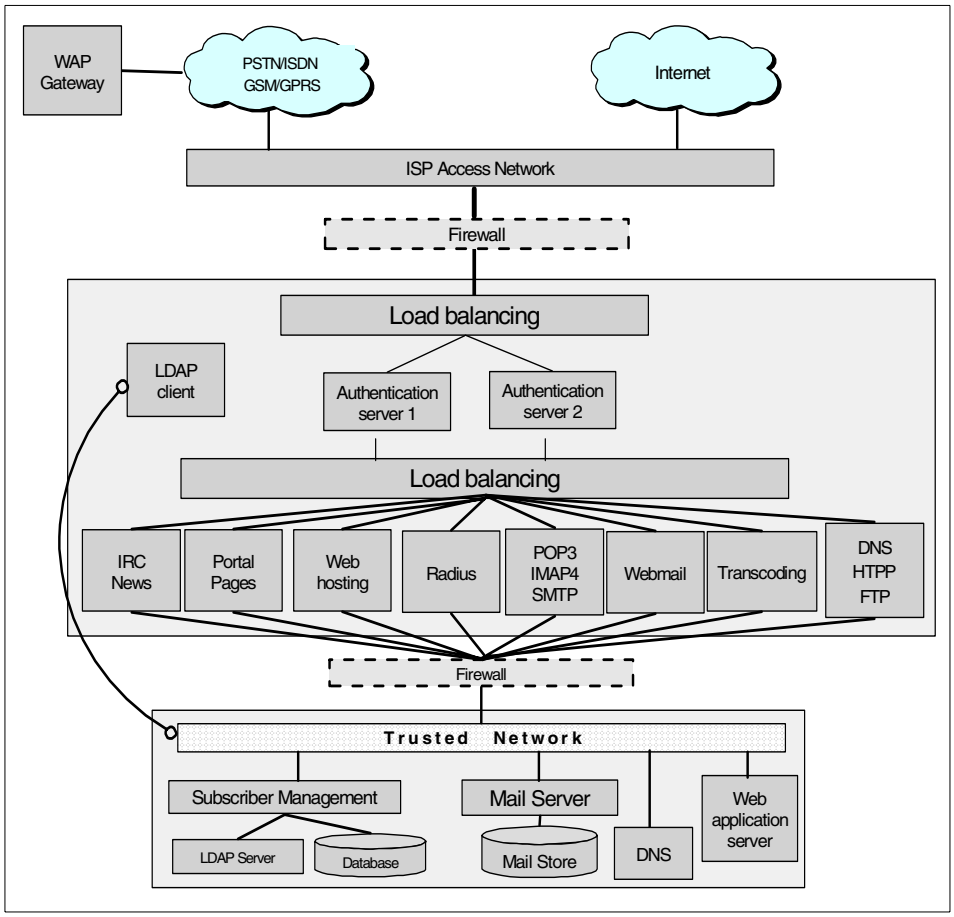


Figure 65. Logical architecture for the enhanced e-business services model

There are new components added to the DMZ:

WAP Gateway External component that provides access to the ISP via the WAP protocol for pervasive devices. This component is outside the SP infrastructure.

Transcoding Component that provides the translation of the services to HTML or WML code, depending on the target device that sends the request.

IRC services Internet Relay Chat (IRC) provides a mechanism to communicate, in real time, with people from all over the world.

News groups InterNet News (INN) is a set of protocols for exchanging messages in a decentralized network of news servers. Each individual news server locally stores all the articles it has received for a given news group.

Security is managed, as in the previous model.

5.7.2 Physical architecture

From the point of view of the physical architecture, this model does not have many differences from the managed e-business services model. The only differences are in the DMZ zone and in some external devices, as shown in Figure 66.

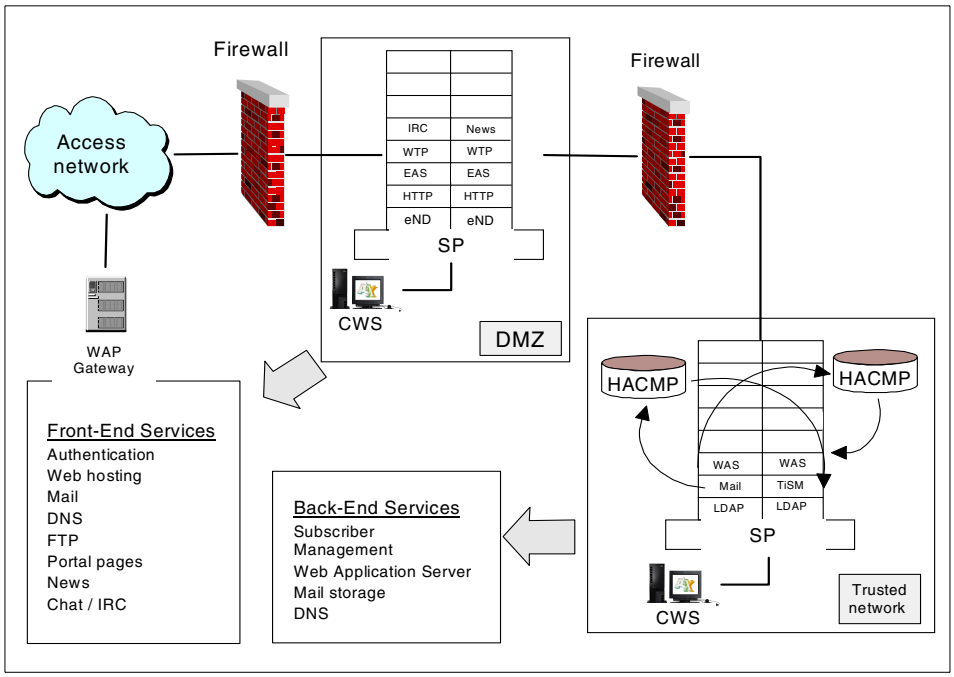


Figure 66. Physical architecture for the enhanced e-business services model

DMZ frame

We added two new nodes to host a new transcoding service based on the WTP (Web Transcoding Publisher) product. The nodes are managed with the eND feature to ensure load balancing and high availability.

There is one node providing news service using the NNTP protocol and another node dedicated to provide IRC services. These services are not

configured with high availability because they are not critical to the functionality of the ISP. If the ISP loses these services, the core services will not be affected at all.

Outside of the SP infrastructure, we added a WAP Gateway server to provide the ISP connectivity with pervasive devices.

The security configuration is composed of two firewalls that separate the DMZ frame and the Trusted network frame, ensuring security in the back end.

5.7.3 Further development

The three models provide a general overview of the ISP's physical configurations. These configurations are not suitable for a production environment, as some tailoring is still required. Configuration for a production environment depends on the ISP's functionality and requirements. Deployment and hardware infrastructure are based on these requirements. Products that are used to implement the ISP platform require specifications geared to their respective features and capabilities.

During the sample descriptions, we did not go into the products' specific details.

Chapter 6. Sample implementation

This chapter provides a brief overview on how we implemented the ISP components into the RS/6000 SP environment. This sample installation will be based on Section 5.2, “Market visibility model” on page 199 and Section 5.3, “Enhanced market visibility model” on page 202.

Although it seems to be basic (nothing new) for some readers, we will start this chapter with basic tasks, such as node installation, setup, and customization steps. These steps will include setup of time server, file collections, and globalized NFS automounted directories. These NFS automounted directories create the basis from which basic ISP services and applications can be easily deployed. On the other hand, a well planned basic setup will ensure better operation of the ISP in terms of scalability, flexibility, and maintainability within the RS/6000 SP environment.

Later, we will introduce DNS, FTP, and mail services as part of the basic preparation for ISP Services. We will simplify the ISP setup by using IBM Network Dispatcher and IBM HTTP Server with LDAP authentication, based on IBM SecureWay Directory Server with DB2 Database connectivity.

We may not be able to achieve every requirement that an ISP may require. For example:

- Trusted Computing Base (TCB)
- High Availability Cluster Multi-Processing (HACMP)
- Distributed Computing Environment (DCE) or Distributed Filesystem (DFS)
- Special licensed software products or free available ones
- Detailed security requirements

Although we try to show the integration by using step by step scenarios, we assume the RS/6000 SP and Control Workstation (CWS) as installed. For details on the RS/6000 SP installation, refer to the *Parallel System Support Programs for AIX Installation and Migration Guide*, GA22-7347.

Furthermore, we are not describing the network elements needed to create the ISP network infrastructure. For example:

- Firewall products and implementations
- Router and bridge products

6.1 Basic sample implementation layout

The following sections describe the layout used within the ITSO Poughkeepsie environment. The sample implementation has been performed on a one frame RS/6000 SP with six thin nodes.

6.1.1 Network layout

Figure 67 shows the sample Internet Protocol (IP) Addressing layout used for our sample installation. The physical type of the network used, for example, Ethernet or ATM, did not have any impact on the IP addressing or the sample hosting services. Furthermore, we took advantage of aliasing IP addresses to network interfaces. This enabled ISP services that are normally bound to a hostname to be deployed regardless of the physical system; they are therefore not listed within Figure 67.

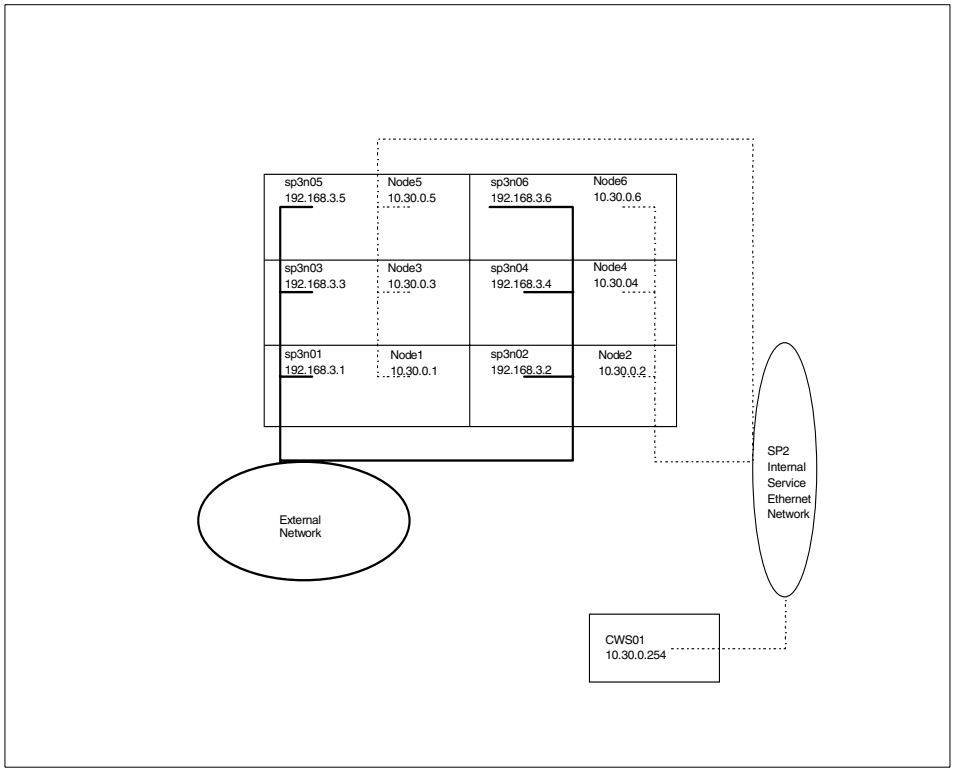


Figure 67. Sample ISP network layout

6.1.2 RS/6000 SP filesystem layout

Figure 68 shows the logical filesystem layout for our sample installation. We attached eight Serial Storage Architecture (SSA) Disks to the Control Workstation (CWS), four disks to node4, and four disks to node6. The CWS and node4 attachments were configured as SSA RAID Level 5 Drives with one spare disk each. In order for there to have been a general access method from all nodes, we used the Network File System (NFS) automounter shipped with Parallel System Support Programs (PSSP).

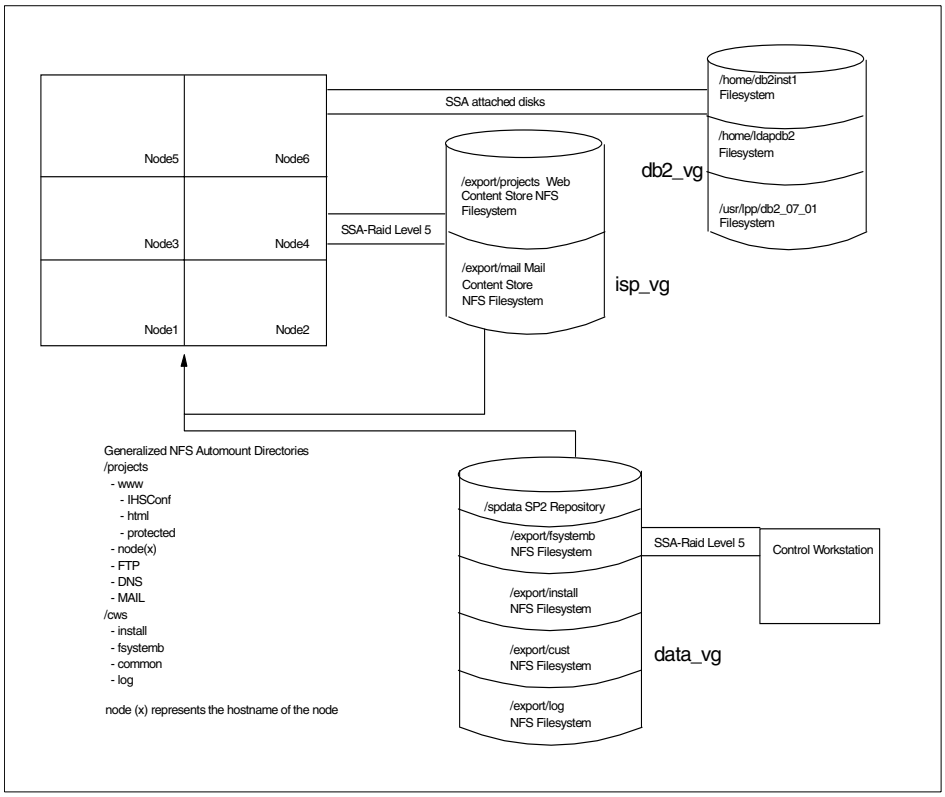


Figure 68. RS/6000 SP filesystem layout

6.1.3 Application layout

Figure 69 on page 222 shows the application layout for our sample installation. All installed applications are part of AIX V4.3 or WebSphere (TM) EveryPlace Server for Multiplatforms products.

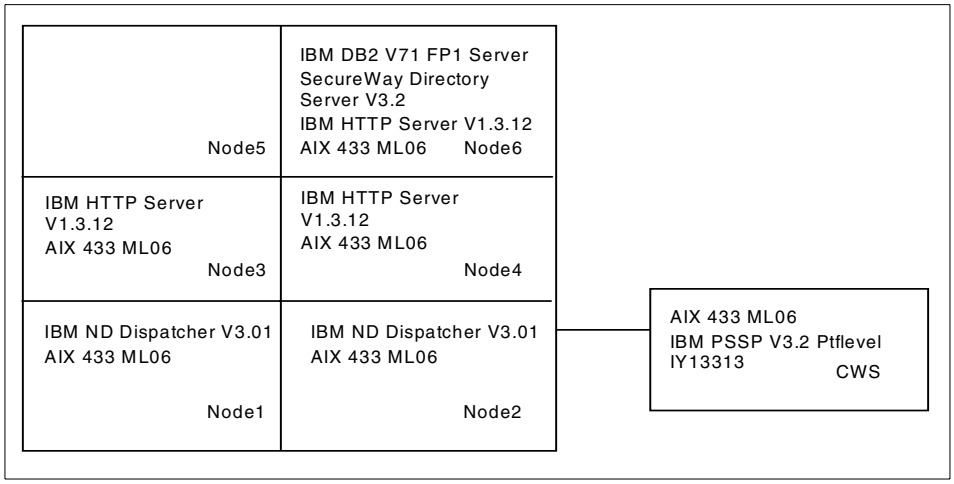


Figure 69. Sample RS/6000 SP application layout

6.2 Preparing the control workstation

Before installing any node or product, we needed to tailor the CWS to ensure proper operation of the whole SP. The CWS has the capability to install, customize, and administrate applications deployed on any node.

6.2.1 Changing root users home directory

Changing the root user's home directory from the file systems root mount point `(/)` to /home/root prevents problems caused by the operating system's usage of the user's home directory ($HOME) or careless use of system commands within $HOME. For example:

- Temporary files, such as .sh_history, smit.log and smit.script, were intentionally placed in $HOME.
- The command `cd` without options causes the shell to return to $HOME. Sometime this causes problems when using the AIX command `rm`.

We used the following command sequence to change to change the root users home directory to /home/root:

```
<cws01>#chuser home=/home/root root
<cws01>#mkdir /home/root
<cws01>#chmod 700 /home/root
<cws01>#chown root.system /home/root
```

Note

This should be done before setting up authentication services. If it has already done, you need to recreate the authentication database, as described in Section 3.3.1 of *PSSP V3 Survival Guide,* SG24-5344.

6.2.2 Setting up the time server

Although most of the services and applications do not rely on exact time synchronization, usage of an exact time base is reasonable. Thus, we defined the CWS as the time provider for all nodes that were using Network Time Protocol (NTP). The CWS itself collected the time from another timeserver.

For details on NTP, refer to:

`http://www.eecis.udel.edu/~ntp.`

The use of NTP within the SP can be generally set during the initial setup of the SP site environment, or by using the following command:

```
<cws01>#smitty site_env_dialog
or
<cws01>#spsitenv ntp_config=consensus
```

We customized the NTP settings within /etc/ntp.conf file on the CWS and chose to use the time servers:

```
server ntp0.cornell.edu version 2
server ntp.cmr.gov version 2
```

Note

You should always notify the owner of a time server that your are going to reference his time server. A time server may have open access, but you will be not be notified about any changes or outages if you are not registered.

Connectivity to the time server was checked when we used the following AIX command:

```
<cws01>#xntpdc -c peers ntp0.cornell.edu
```

NTP will synchronize the time in small steps.

Another way to synchronize the system time using an external time provider can be achieved using the command `ntpdate`. This command changes the system time immediately on execution. This can be done on a scheduled basis using a crontab entry:

```
15 22 * * * /usr/sbin/ntpdate -s -u ntp0.cornell.edu
```

We decided to use the crontab entry on the CWS to synchronize the system time. It does not have any effect on the RS/6000 SP Site environment NTP settings. That means the CWS synchronizes its system time with the time provider at 10:15 PM every day. The nodes synchronize their system time with the CWS using the NTP RS/6000 SP default settings.

6.2.3 Tailoring firstboot.cust, tuning.cust and script.cust

PSSP provides the opportunity to run three different customer-supplied scripts to customize node setup, as shown in Table 2 on page 225. We used the scripts as a focal point for general changes that affected the operating system environment during the initial installation of the master install image, as well as for general changes to were performed during scheduled maintenance.

Table 2. PSSP customization scripts

Script Name	executed during	purpose
script.cust	install, customize boot	Customizations that require a reboot of the system
firstboot.cust	install, migrate boot	Customizations that do not require a reboot of the system
tuning.cust	every boot	Network parameters

We copied the PSSP Scripts from /spdata/sys1/install/aix433/spot/spot_aix433/usr/lpp/ssp/samples to the directory /tftpboot. We then edited the files to make the changes that fit our needs:

- script.cust file:

```
##################
# Following lines have been added for redbook SG24-6025
#
#change the primary sysdump lv, enable auto boot after crash
mklv -y'sysdump_lv' -t'sysdump' rootvg 15
sysdumpdev -P -p'/dev/sysdump_lv'
#change the system to enable automatic ipl after crash
chdev -l sys0 -a autorestart='true'
#let the system calculate the maximum network buffer size
chdev -l sys0 -a maxmbuf='0'
#set the maximum user procs to 250
chdev -l sys0 -a maxuproc='250'
#set the bootlog file to 48K
alog -C -t boot -f /var/adm/ras/bootlog -s 48000 -w 1
#
/usr/bin/tftp -o  /etc/netsvc.conf $SERVER /tftpboot/.netsvc.conf
/usr/bin/tftp -o  /etc/passwd $SERVER /tftpboot/.passwd.node
/usr/bin/tftp -o  /etc/group $SERVER /tftpboot/.group.node
#
# End of additions for redbook SG24-6025
##################
```

- firstboot.cust:

```
##################
# Following lines have been added for redbook SG24-6025
#
HOSTNAME=$(hostname)
/usr/sbin/mkitab other:2:once:"/cws/cust/${HOSTNAME}/etc/rc.include
```

```
1 1>/dev/null 2>&1"
#
/usr/lpp/ssp/rcmd/bin/rcp $SERVER:/export/cust/${HOSTNAME}
1 /etc/aliases /etc/aliases
#
cp /etc/rc.tcpip /etc/rc.tcpip.org.${HOSTNAME}
/usr/lpp/ssp/rcmd/bin/rcp $SERVER:/export/cust/${HOSTNAME}
1 /etc/rc.tcpip /etc/rc.tcpip
chmod 755 /etc/rc.tcpip
#
cp /etc/services /etc/services.org
/usr/lpp/ssp/rcmd/bin/rcp $SERVER:/export/cust/${HOSTNAME}/etc/services
1 /etc/services
#
cp /etc/inetd.conf /etc/inetd.conf.org
/usr/lpp/ssp/rcmd/bin/rcp $SERVER:/export/cust/${HOSTNAME}
1 /etc/inetd.conf /etc/inetd.conf
#
cp /etc/syslog.conf /etc/syslog.conf.org
/usr/lpp/ssp/rcmd/bin/rcp $SERVER:/export/cust/${HOSTNAME}
1 /etc/syslog.conf /etc/syslog.conf
#
#create the needed User Home Directories
GROUP="www"
#
for USER in db2inst1 \
               db2fenc1 \
               db2as1   \
               ldapdb2 \
               ftp \
               anonymou\
               ldap     \
               ihsadmin
do
   mkdir /home/${USER}
   chmod 750 /home/${USER}
   chown ${USER}.${GROUP} /home/${USER}
done
#
/usr/lpp/ssp/rcmd/bin/rcp $SERVER:/tftpboot/.profile.node
1  /home/root/.profile
/usr/lpp/ssp/rcmd/bin/rcp $SERVER:/tftpboot/.kshrc.node
1  /home/root/.kshrc
chmod 700 /home/root/.kshrc
chmod 700 /home/root/.profile
#
/usr/lpp/ssp/rcmd/bin/rcp $SERVER:/tftpboot/.profile.etc
```

```
1  /etc/profile
/usr/lpp/ssp/rcmd/bin/rcp $SERVER:/tftpboot/.environment.etc
1  /etc/environment
/usr/lpp/ssp/rcmd/bin/rcp $SERVER:/tftpboot/.motd.etc
1  /etc/motd
#
/usr/lpp/ssp/rcmd/bin/rcp $SERVER:/tftpboot/.hosts.etc
1  /etc/hosts
/usr/lpp/ssp/rcmd/bin/rcp $SERVER:/tftpboot/.resolv.etc
1  /etc/resolv.conf
/usr/lpp/ssp/rcmd/bin/rcp $SERVER:/tftpboot/.netsvc.etc
1  /etc/netsvc.conf
#
mkdir /usr/local
usr/sbin/mknfsmnt -f /usr/local -d /usr/local -h $SERVER
1  -n -I -A -t ro -w bg -S
#
/usr/dt/bin/dtconfig  '-d'
#
# End of aditions for  redbook SG24-6025
###################
```

1 These lines have been split for redbook printing purposes. In the actual file, they are on a single line.

- tuning.cust:

  Nothing relevant for the sample installation.

6.2.4 Creating the common tools data repository on the CWS

Operating an ISP environment sometimes requires tools for daily work that are not part of an AIX standard tool set. The tools source may come from:

- Freeware download servers
- Self-written common procedures

Thus, UNIX system administrators usually use the directory /usr/local for the common tools storage area. We created this storage, within the CWS SSA Raid Level 5 storage devices, in a volume group named data_vg, as shown in Figure 68 on page 221

The following commands were used to create a common tools data repository on the CWS:

```
<cws01>#mklv -y usr_local_lv data_vg 1
<cws01>#crfs -v jfs -d usr_local_lv -m /usr/local -A yes -p rw
1  -t no -a frag=4096 -a nbpi=4096 -a ag=8
<cws01>#chmod 755 /usr/local
<cws01>#chown root.system /usr/local
<cws01>#mount all
<cws01>#chmod 755 /usr/local
<cws01>#chown root.system /usr/local
```

1 These lines have been split for redbook printing purposes. In the actual command sequence, they are on a single line.

Table 3 shows the tools installed from the www.us.bull.com freeware download site into the /usr/local directory.

Table 3. List of installed freeware from www.us.bull.com site

Fileset Name	Level freeware	Brief description
freeware.COPS.rte	1.0.4.1	Security Checker
freeware.unzip.rte	5.41.0.0	Unzip, archiving and compression tool
freeware.monitor.rte	2.1.7.4	Performance Monitor Program
freeware.lsof.rte	4.50.0.0	List Open files
freeware.gnu.gzip.rte	1.2.4.0	GZIP compression and decompression tools
freeware.aix.tools.rte	1.5.2.0	Useful utilities, like which lpp
freeware.wu-ftpd.rte	2.6.1.0	ftp daemon
freeware.sendmail.rte	8.11.1.0	sendmail
freeware.bind.rte	8.2.2.5	DNS

We created an entry in the /etc/exports file so that all the nodes could access the newly created filesystem in read only mode:

```
/usr/local -root,ro cws01:node1:node2:node3:node4:node5:node6
```

To make the changes take effect, we executed the following command:

```
<cws01>#exportfs -va
```

6.2.5 Creating the common full system backup data repository

Backup and recovery strategies are key elements of daily operations. The RS/6000 SP environment allows you to easily install or recover nodes from a full system backup when using the PSSP build in Network Install Manager (NIM). We defined the CWS to hold the filesystems on which scheduled nodes full system backups would be stored.

The following commands were used to create the full system data repository for backups:

```
<cws01>#mklv -y fsystemb_lv data_vg 1
<cws01>#crfs -v jfs -d fsystemb_lv -m /export/fsystemb -A yes
1  -p rw -t no -a frag=4096 -a nbpi=4096 -a ag=8
<cws01>#chmod 755 /export/fsystemb
<cws01>#chown root.system /export/fsystemb
<cws01>#mount all
<cws01>#chmod 755 /export/fsystemb
<cws01>#chown root.system /export/fsystemb
```

1 These lines have been split for redbook printing purposes. In the actual command sequence, they are on a single line.

We created a subdirectory for each SP node so that we could use simple generalized scripts to create and identify the full system backup.

The following commands created subdirectories for each SP node:

```
<cws01>#cd /exports/fsystemb
<cws01>#mkdir node1 node2 node3 node4 node5 node6
<cws01>#chmod 775 node*
<cws01>#chown root.system node*
```

These directories were mounted using NFS, so we created an entry in the /etc/exports file so that all the nodes could access the newly created filesystem in read/write mode:

```
/exports/fsystemb -root,rw node1:node2:node3:node4:node5:node6
```

To make the changes take effect, we executed the following command:

```
<cws01>#exportfs -va
```

6.2.6 Creating the common configuration data repository

The common configuration data repository was used as the centralized focal point for storing the common configuration files for deployment of applications (shell scripts) or nodes. Furthermore, we stored tailored node dependant configuration files (startup scripts) during the initial boot.

The following commands created a centralized area on the CWS for storing common configuration files that can be used for deployment of applications or nodes:

```
<cws01>#mklv -y cust_lv data_vg 1
<cws01>#crfs -v jfs -d cust_lv -m /export/cust -A yes -p rw
1  -t no -a frag=4096 -a nbpi=4096 -a ag=8
<cws01>#chmod 755 /export/cust
<cws01>#chown root.system /export/cust
<cws01>#mount all
<cws01>#chmod 755 /export/cust
<cws01>#chown root.system /export/cust
```

1 These lines have been split for redbook printing purposes. In the actual command sequence, they are on a single line.

We created the subdirectory /export/cust/common to store common scripts for all services. We also created, for each node, a subdirectory where basic node configuration files could be stored, for example, rc.tcpip, inetd.conf. The following commands show how these tasks were accomplished:

```
<cws01>#mkdir /export/cust/common
<cws01>#mkdir -p cws01/etc node1/etc node2/etc node3/etc
1 node4/etc node5/etc node6/etc
<cws01>#chmod -R 755 /export/cust/*
```

1 These lines have been split for redbook printing purposes. In the actual command sequence, they are on a single line.

We create an entry in the /etc/exports file that allowed all the nodes too access the newly created filesystem in read only mode:

```
/exports/cust -root,ro cws01:node1:node2:node3:node4:node5:node6
```

For the changes to take effect, we executed the command:

```
<cws01>#exportfs -va
```

6.2.7 Creating the common installables data repository

The common installables data repository was used as the centralized focal point for storing all software products that were installed on single or multiple nodes. The following commands show you how this task was accomplished:

```
<cws01>#mklv -y install_lv data_vg 1
<cws01>#crfs -v jfs -d install_lv -m /export/install -A yes
1  -p rw -t no -a frag=4096 -a nbpi=4096 -a ag=8
<cws01>#chmod 755 /export/install
<cws01>#chown root.system /export/install
<cws01>#mount all
<cws01>#chmod 755 /export/install
<cws01>#chown root.system /export/install
```

1 These lines have been split for redbook printing purposes. In the actual command sequence, they are on a single line.

We created an entry in the /etc/exports file so that all the nodes could access the new created filesystem in read only mode:

```
/exports/install -root,ro cws01:node1:node2:node3:node4:node5:node6
```

For the changes to take effect, we executed the command:

```
<cws01>#exportfs -va
```

6.2.8 Creating the common log data repository

The common configuration repository was used as the centralized focal point for storing log files written by applications during the installation of the applications. This enabled us to examine the logfiles without logging onto the node itself.

```
<cws01>#mklv -y log_lv data_vg 1
<cws01>#crfs -v jfs -d log_lv -m /export/log -A yes -p rw -t no
1 -a frag=4096 -a nbpi=4096 -a ag=8
<cws01>#chmod 775 /export/log
<cws01>#chown root.system /export/log
<cws01>#mount all
<cws01>#chmod 775 /export/log
<cws01>#chown root.system /export/log
```

1 These lines have been split for redbook printing purposes. In the actual command sequence, they are on a single line.

We are going to created subdirectories for each SP node and the CWS itself to store logfiles. The directory names are based on the system internal SP Ethernet hostname. The following commands show you how this task was accomplished:

```
<cws01>#cd /exports/log
<cws01>#mkdir cws01 node1 node2 node3 node4 node5 node6
<cws01>#chmod 775 node*
<cws01>#chown root.system cws* node*
```

We created an entry in the /etc/exports file so that all nodes could access the new created filesystem in read/write mode:

```
/exports/log -root,rw cws01:node1:node2:node3:node4:node5:node6
```

For the changes to take effect, we executed the following command:

```
<cws01>#exportfs -va
```

6.2.9 Basic security setup

Security and setup settings are spread over a wide area of tasks, beginning with the hardware layer up to the high level software security. There are several published sources regarding system security, for example, the redbook *AIX 4.3 Elements of Security: Effective and Efficient Implementation*, SG24-5962.

In this section, we are trying show how we avoided selected TCP/IP services from being started or made accessible on our SP nodes. PSSP itself uses

internal authentication methods during execution of PSSP commands, such as `dsh` or `spmon`, and are not affected by the changes described.

With the AIX Base Operating System (BOS) installed, several initial TCP/IP services are enabled by the inetd daemon or automatically started during the execution of rc.tcpip, which may make the system vulnerable. Thus, we updated the corresponding configuration files rc.tcpip and inetd.conf and replaced the original files with the customized ones during execution of the script firstboot.cust. See Section 6.2.3, “Tailoring firstboot.cust, tuning.cust and script.cust” on page 224 for more details about the firstboot.cust script used for the sample implementation.

The scripts were placed on the nodes’ generalized configuration directory (/cws/cust/<nodename>/etc) in order to customize each node separately. The contents of the files used can be found in Appendix B, “Sample configuration files” on page 287.

6.2.10 File collections for /etc/passwd and /etc/group updates

To have the same User ID and Group ID layout spread over all the nodes and the CWS, we used the PSSP built-in feature called *file collections*. Deployment of this feature can be set during the initial setup of the SP Site Environment or by using the PSSP command `spsitenv`:

```
<cws01>#smitty site_env_dialog
or
<cws01>#spsitenv filecoll_config=true
```

We know by now that some applications do need special User IDs and Group IDs to be installed. Thus, we created these user and group IDs once on the CWS, and used the PSSP file collections to push these files to each node.

Using the file collection mechanism, we avoided any subsequent problems when cloning server services from one node to another.

Table 4 shows the additional user names and IDs needed for installing and running the applications that were installed for the sample implementation.

Table 4. Additional User IDs

User Name	Used by application
db2inst1	IBM DB2 Default Instance User
db2fenc	IBM DB2 Fenced User

User Name	Used by application
db2as1	IBM DB2 Admin User
ihsadmin	IBM HTTP Server Admin User
ldapdb2	IBM LDAP DB2 Instance User
ldap	IBM LDAP User ID
ftp	FTP Server
anonymous	FTP Server

Table 5 shows additional group names needed for installing and running applications on the nodes.

Table 5. Additional Group IDs

Group Name	Used by application	Member
dbsysadm	IBM DB2	root, db2as1, ldapdb2, ldap
ldap	IBM LDAP	ldap, ldapdb2
www	Dummy Group	All as primary Group

We added the user and groups, as described in Table 4 and Table 5, to the CWS using the AIX commands `mkgroup` and `mkuser`. The group IDs and user IDs used may vary on other installations. It is not mandatory to use the same values we used. The following commands show how you can accomplish this task:

```
<cws01>#mkgroup id=400 wwww
<cws01>#mkuser home=/home/db2inst1 id=400 pgrp=www db2inst1
```

The previous commands added the following entries to the files /etc/passwd and /etc/group on the CWS:

- /etc/passwd

```
db2inst1:!:400:400::/home/db2inst1:/bin/ksh
db2fenc1:!:401:400::/home/db2fenc1:/bin/ksh
db2as1:!:402:400::/home/db2as1:/bin/ksh
ldapdb2:!:403:400::/home/ldapdb2:/bin/ksh
ihsadmin:!:404:400::/home/ihsadmin:/bin/ksh
ldap:!:405:402::/home/ldap:/bin/ksh
ftp:*:300:1::/home/ftp:/usr/bin/ksh
anonymou:*:301:1::/home/ftp:/usr/bin/ksh
```

- /etc/group

```
www:!:400:db2inst1,db2fenc1,db2as1,ldapdb2
dbsysadm:!:401:root,db2inst1,db2fenc1,db2as1,ldapdb2,ldap
ldap:!:402:ldap,ldapdb2
```

6.2.11 Tailoring the NFS automounter

Automounter is a facility used to manage the mounting activity of a file system. When you access a file or directory under automounter control, the automounter transparently mounts the required file system. When there has been no activity on that file system for some pre-determined amount of time, the automounter unmounts the file system.

We used the RS/6000 SP automounter implementation in order to have a single view of all the nodes that was installed without having to handle different local directories and path names.

In general, this feature can be set during the initial setup of the SP Site Environment or by using the PSSP command `spsitenv`:

```
<cws01>#smitty site_env_dialog
or
<cws01>#spsitenv amd_config=true
```

To enable global access to the common NFS shared automounter repository, we added the following lines to the /etc/auto.master file on the CWS:

```
/cws  /etc/auto/maps/auto.cws
/projects /etc/auto/maps/auto.isp
```

These two configuration files allowed the automounter to resolve the real mount points for directories within the /cws and /projects directory trees.

We created the referenced automounter map files in the directory /etc/auto/maps/ on the CWS and added the following lines to:

- /etc/auto/maps/auto.cws

```
install      cws01:/export/install
fsystemb     cws01:/export/fsystemb
cust         cws01:/export/cust
log          cws01:/export/log
```

- /etc/auto/maps/auto.isp

```
node1 node4:/export/projects/node1
node2 node4:/export/projects/node2
node3 node4:/export/projects/node3
node4 node4:/export/projects/node4
node5 node4:/export/projects/node5
node6 node4:/export/projects/node6
```

Because we decided to use the RS/6000 SP file collections, we used this mechanism to distribute the new NFS automounter configuration files to all the nodes too. We did this by adding the following lines to the end of the file /var/sysman/sup/user.admin/list:

```
execute /etc/amd/refresh_amd (./etc/auto/maps/auto.isp)
execute /etc/amd/refresh_amd (./etc/auto/maps/auto.cws)
```

The CWS also runs the automounter, thus we can use the generalized mount points for administration tasks too. To achieve this, we created the directory /export/cust/cws01 using following steps on the CWS:

```
<cws01>#mkdir /export/cust/cws01
<cws01>#chown root.system /export/cust/cws01
```

We copied the NFS automounter configuration files into the /export/cust/cws01 directory in order to have them available when using the globalized mount point /cws/cust/cws01:

```
<cws01>#cd /export/cust/cws01
<cws01>#cp /etc/auto.master master.auto
<cws01>#cp /etc/auto/maps/auto.isp auto.isp
<cws01>#cp /etc/auto/maps/auto.cws auto.cws
<cws01>#cp /var/sysman/sup/user.admin/list sup.admin.list
```

All changes to these files can be performed within this directory. To update the CWS and all of the nodes' NFS Automounter configurations, we created a simple shell script called AMD_entry.ksh and placed it in the common code directory /cws/cust/common.

This simple script updates all the configuration files and nodes, if it is executed on the CWS. If it is executed on a node, the file collection update is performed after refreshing of the automounter daemon. The script is as follows:

```
#AMD_entry.ksh script for redbook SG24-6025
#
HOST=$(hostname)
CUST="/cws/cust"
#
if [ ! -a /etc/ssp/server_hostname ]
then
# assume to be on the CWS
 echo "*** we are working on CWS ***"
 cd  ${CUST}/${HOST}
 cp master.cws /etc/auto.master
 cp auto.* /etc/auto/maps/
 cp sup.admin.list /var/sysman/sup/user.admin/list
 exportfs -va
 stopsrc -g autofs
 /etc/auto/startauto
 echo "*** starting dsh -i *** on the nodes out of ${HOME}/WCOLL"
 dsh -i /var/sysman/supper update  user.admin
 dsh -i stopsrc -g autofs
 dsh -i /etc/auto/startauto
else
# assume to be on Node
 echo "*** we are working on a Node ***"
 /var/sysman/supper update  user.admin
 stopsrc -g autofs
 /etc/auto/startauto
fi
#
exit 0
```

6.2.12 Update the working collective file WCOLL

The working collective is used to parse each host to the distributed shell (dsh), where specific commands should be executed. We set the environment variable WCOLL to /home/root/WCOLL by default. The working collective file should have one host name per line; blank lines and comment lines beginning with a # in front of the line are ignored.

Later on, we added each node as it was installed and deployed in the production environment. We used the following command to dynamically update the working collective file case by case:

```
<cws01>#SDRGetObjects -G host_responds host_responds==1
1 node_number | grep -v node_number > $HOME/WCOLL
```

1 These lines have been split for redbook printing purposes. In the actual command sequence, they are on a single line.

6.2.13 Creating the first master install image

One of the advantages of using the RS/6000 SP is its built-in Network Install Manager (NIM) deployment tool. To achieve the minimal post installation tasks on each node, we installed one node with all the needed AIX components and created a full system backup. Later on, we used this image, called the master install image, for cloning purposes.

We defined node1 to be used as the master image creation node. We used the following steps to enable the Base Operating System (BOS) installation from the CWS:

```
<cws01>#spbootins -r install 1 1 1
<cws01>#nodecond 1 1
```

While the node was installing the AIX BOS, we created a bundle file with all the additional required AIX components to enable easy installation of all AIX related Licensed Software Products (LPP). The file has been named ISP.bundle and was placed into directory /tftpboot on the CWS. The ISP bundle file is as follows:

```
#ISP.bundel simple Bundle file for redbook SG24-6025
# Accounting Services
bos.acct
# BOS Application Development Tools
bos.adt.include
bos.adt.prof
# AIXwindows Application Development Tools
X11.adt.lib
X11.adt.motif
# AIXwindows Motif Runtime Environment
X11.motif.lib
X11.motif.mwm
# AIXwindows Runtime Environment
X11.base.lib
X11.base.rte
X11.compat.lib.X11R5
# AIX Desktop runtime libraries
X11.Dt.rte
X11.Dt.lib
# AIX pthreads
bos.rte.libpthreads
```

```
# Java Runtime
Java.rte.bin
Java.rte.classes
Java.rte.lib
```

After the initial AIX BOS installation finished, we installed the missing AIX LPPs on node1, as follows:

```
<cws01>#dsh -w node1 mount cws01:/spdata/sys1/install/aix4336/
1 lppsource /mnt
<cws01>#rcp /tftpboot/ISP.bundle node1:/usr/sys/inst.data/
1 user_bundles/ISP.bnd
<cws01>#dsh -w node1 /usr/lib/instl/sm_inst installp_cmd -a
1 -Q -d '/mnt/' -b 'ISP' -f '_all_licensed'  '-c' '-N' '-g'
1 '-X'  '-G'
<cws01>#dsh -w node1 umount /mnt
```

1 These lines have been split for redbook printing purposes. In the actual command sequence, they are on a single line.

All further needed customizations were already performed by firstboot.cust, script.cust and tuning.cust, which are located within directory /tftpboot in the CWS, as described in Section 6.2.3, “Tailoring firstboot.cust, tuning.cust and script.cust” on page 224. To keep us capable of installing software without booting the node in either customize or migrate mode, we did not include this step in script.cust.

We wanted to ensure that latest updates for the NFS automounter configuration are pulled from the CWS and activated on the node so that we could use the globalized directory structure. To do so, we updated the node1 NFS automounter configuration using the following:

```
<cws01>#dsh -w node1 /var/sysman/supper update  user.admin
<cws01>#dsh -w node1 stopsrc -g autofs
<cws01>#dsh -w node1 /etc/auto/startauto
```

We were then ready to create the full system backup from node1, which we used as the source for the install on all the other nodes. The following command is used to accomplish this task:

```
<cws01>#dsh -w node1 mksysb -i /cws/fsystemb/master.fsystemb
```

6.2.14 Tailoring the master install image

During the installation of other applications, we found that Java AIX fileset was not installed, so it did not become part of the master install image, although it is required within our bundle file. The problem was solved by placing this fileset into the lppsource (/spdata/sys1/install/aix4336/lppsource) on the CWS. We ran the `installp` command, using the command `dsh -i`, to install the missing filesets on all nodes listed within the working file collection:

```
<cws01>#dsh -i rcp cws01:/tftpboot/ISP.bundle /usr/sys/
1  inst.data/user_bundles/ISP.bnd

<cws01>#dsh -i mount cws01:/spdata/sys1/install/aix4336/
1 lppsource /mnt
<cws01>#dsh -i /usr/lib/instl/sm_inst installp_cmd -a -Q
1  -d '/mnt/' -b 'ISP' -f '_all_licensed'  '-c' '-N' '-g' '-X'
1  '-G' -e /tmp/install.log
<cws01>#dsh -i umount /mnt
```

1 These lines have been split for redbook printing purposes. In the actual command sequence, they are on a single line.

At this point, we replaced the old default master install image with the recent one:

```
<cws01>#dsh -w node1 mksysb -i /cws/fsystemb/master.fsystemb
```

Installed AIX or LPP updates can be applied anytime using a single command issued from the CWS:

```
<cws01>#dsh -i /usr/lib/instl/sm_inst installp_cmd -c -f'all'
1  '-g' '-X'
```

1 These lines have been split for redbook printing purposes. In the actual command sequence, they are on a single line.

6.2.15 Setting additional node network IP addresses

Before we installed all the nodes from the master install image, we needed to add the nodes' additional network interfaces with the correct IP addresses into the PSSP System Data Repository (SDR). As Figure 67 on page 220 shows, we used IP addresses in the range 192.168.3.1 to 192.168.3.26 for accessing the external network.

The SDR data repository enables us to install nodes without having to take care of subsequent network interface setups, because this will be automatically performed for all the network adapters stored within the SDR. This step was done on the CWS using SP node database information or the command `spadaptrs`. We show here how we used the SP node database information dialogue, which we entered using the SMIT fastpath `smit add_adapt_dialog`:

```
 Additional Adapter Database Information

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                    [Entry Fields]
  Start Frame                                      [1]
  Start Slot                                       [1]
  Node Count                                       [6]

  OR

  Node Group                                       []

  OR

  Node List                                        []

* Adapter Name                                     [en1]
* Starting Node's IP Address or Hostname           [192.168.3.1]
* Netmask                                          [255.255.255.0]
  Additional IP Addresses                          []
  Ethernet Adapter Type                            dix
  Duplex
  Ethernet Speed                                   auto
  Token Ring Data Rate
  Skip IP Addresses for Unused Slots?              no
  Enable ARP for the css0 Adapter?                 no
  Use Switch Node Numbers for css0 IP Addresses?   no




F1=Help              F2=Refresh           F3=Cancel            F4=List
F5=Reset             F6=Command           F7=Edit              F8=Image
F9=Shell             F10=Exit             Enter=Do
```

6.2.16 Saving the PSSP data repository

PSSP SDR is the essential data repository from an RS/6000 SP operational point of view, because all basic physical and logical information about the

RS/6000 SP system are stored within. Here we show how we archived the repository:

```
<cws01>#SDRArchive
0025-322  SDRArchive: SDR archive file name is /spdata/sys1/
1  sdr/archives/backup.00307.1455
```

1 These lines have been split for redbook printing purposes. In the actual command and output sequence, they are on a single line.

6.3 Preparing application deployment

In this section, we will show how we installed applications that are needed for basic ISP operations, such as Domain Name Service (DNS), Mail Gateway, and FTP server.

Before we started deploying applications, we needed to set up basic Internet services like Mail and DNS services. With the installation of the general master image, the nodes were prepared to access the globalized directory tree that we used to configure and deploy these services.

6.3.1 Installing all nodes with the master install image

We were ready to install the master boot image and all the other nodes within the RS/6000 SP frame. To keep in line with the standard PSSP setup, we:

1. Copied the master install image into the PSSP default install image directory /spdata/sys1/install/images as file master.1:

```
<cws01>#cd /spdata/sys1/install/images
<cws01>#cp /cws/fsystemb/master.fsystemb master.1
```

2. Changed the volume group information for all uninstalled nodes using the command `spchvgobj` in order to access the new node boot image named master.1:

```
<cws01>#spchvgobj -i master.1 -p PSSP-3.2 1 2 6
or
<cws01>#smitty changevg_dialog
```

3. We then set all the nodes to install using the command `spbootins`:

```
<cws01>#spbootins -r install 1 2 6
or
<cws01>#smitty server_dialog
```

4. Installation of all the nodes was accomplished by using the command `nodecond`:

```
<cws01>#nodecond 1 1&
<cws01>#nodecond 1 2&
<cws01>#nodecond 1 3&
<cws01>#nodecond 1 4&
<cws01>#nodecond 1 5&
<cws01>#nodecond 1 6&
```

The total time used for an initial install of all nodes depends on several factors, which we will not cover in this redbook. For details about the boot and install process, refer to the IBM publication *Parallel System Support Programs for AIX Diagnosis Guide Version 3 Release 2,* GA22-7350.

Installation progress can be checked anytime using the `spmon` or `perspectives` PSSP monitors. The following shows the sample output of the command `spmon -d` after the successful installation of all nodes:

```
<cws01>#spmon -d
5.  Checking nodes
---------------------------------- Frame 1 ----------------------
                          Host     Switch   Key     Env   Front Panel
Slot Node Type  Power Responds Responds Switch  Error LCD/LED
---- ---- ----- ----- -------- -------- ------- ----- ------------
1     1  thin   on     yes      no     normal   no  LEDs are blank
2     2  thin   on     yes      no     normal   no  LEDs are blank
3     3  thin   on     yes      no     normal   no  LEDs are blank
4     4  thin   on     yes      no     normal   no  LEDs are blank
5     5  thin   on     yes      no     normal   no  LEDs are blank
6     6  thin   on     yes      no     normal   no  LEDs are blank
```

6.3.2 Creating the general data repository store

In order to use the generalized NFS automounter mount points, as described in Section 6.1.2, “RS/6000 SP filesystem layout” on page 221, we needed to generate the file systems on the defined node4.

The physical SSA disks were grouped together in one SSA RAID Level 5 disk named hdisk3. We installed the volume group (VG) named isp_vg, and later

on installed, step by step, the needed file systems for basic application deployment on the nodes:

```
<cws01>#dsh -w node4 mkvg -f -y isp_vg -s 32 hdisk3
<cws01>#dsh -w node4 mklv -y projects_lv isp_vg 4
<cws01>#dsh -w node4 crfs -v jfs -d projects_lv -m /export/projects -A ye
1  -p rw -t no -a frag=4096 -a nbpi=4096 -a ag=8
<cws01>#dsh -w node4 chmod 755 /export/projects
<cws01>#dsh -w node4 chown root.system /export/projects
<cws01>#dsh -w node4 mount all
<cws01>#dsh -w node4 chmod 755 /export/projects
<cws01>#dsh -w node4 chown root.system /export/projects
```

1 These lines have been split for redbook printing purposes. In the actual command sequence, they are on a single line.

We created an exports file in the directory /cws/cust/node4/etc:

```
/exports/projects -root,ro cws01:node1:node2:node3:node4:node5:node6
```

The following command shows the export file pushed to node4 and the NFS server updated:

```
<cws01>#dsh -w node4 cp /cws/cust/node4/etc/exports /etc/exports
<cws01>#dsh -w node4 exportfs -va
```

We created a directory entry for every node in advance, in case we needed to store services and node specific configuration files within the globalized directory path /projects/<nodename>:

```
<cws01>#dsh -w node4 mkdir /export/projects/node1
<cws01>#dsh -w node4 mkdir /export/projects/node2
<cws01>#dsh -w node4 mkdir /export/projects/node3
<cws01>#dsh -w node4 mkdir /export/projects/node4
<cws01>#dsh -w node4 mkdir /export/projects/node5
<cws01>#dsh -w node4 mkdir /export/projects/node6
<cws01>#dsh -w node4 chmod 755 /export/projects/node*
```

We did not need to update the NFS automounter configuration file /cws/cust/cws01/auto.isp on the CWS, because the entries have already been made in advance in Section 6.2.11, “Tailoring the NFS automounter” on page 235.

We redistributed the NFS Automounter configuration files using the script AMD_entry.ksh, which is located within /cws/cust/common, as described in Section 6.2.11, “Tailoring the NFS automounter” on page 235:

```
<cws01>#/cws/cust/common/AMD_entry.ksh
*** we are working on CWS ***
exportfs: 1831-187 re-exported /spdata/sys1/install/pssplpp
exportfs: 1831-187 re-exported /export/fsystemb
exportfs: 1831-187 re-exported /export/install
exportfs: 1831-187 re-exported /export/cust
exportfs: 1831-187 re-exported /export/projects
exportfs: 1831-187 re-exported /spdata/sys1/install/pssp/
1  bosinst_data_prompt
exportfs: 1831-187 re-exported /spdata/sys1/install/aix4334
1  /spot/spot_aix4334/usr
exportfs: 1831-187 re-exported /spdata/sys1/install/aix4334/
1  lppsource
0513-044 The automountd Subsystem was requested to stop.
*** starting dsh -i *** on the nodes out of /home/root/WCOLL
Working collective:
node1
node2
>>> /var/sysman/supper update user.admin
node1: Updating collection user.admin from server cws01
node1: File Changes:  3 updated, 0 removed, 0 errors.
node2: Updating collection user.admin from server cws01.
node2: File Changes:  3 updated, 0 removed, 0 errors.
```

1 These lines have been split for redbook printing purposes. In the actual command and output sequence, they are on a single line.

6.3.3 Domain name system configuration

To implement the Domain Name System (DNS), we used the standard AIX named distribution installed in the AIX fileset bos.net.tcp.server. To achieve ease of use, we used the h2n (host-2-named) Perl script to convert the standard hosts file format into the named configuration database files.

We downloaded the h2n utility from the Internet Web site `http://www.dns.net/dnsrd/tools.html` and placed it into the directory /cws/cust/common for later use.

We decided to use Berkeley Internet Named Daemon (BIND) Version 8 style definition files. By default, the current AIX V4.3 distribution uses the BIND Version 4 named daemon. Thus, we needed to prepare the node to allow the DNS server to run with the higher version of BIND by using the following:

```
<cws01>#dsh -w node1 ln -sf /usr/sbin/named8 /usr/sbin/named
<cws01>#dsh -w node1 ln -sf /usr/sbin/named8-xfer
/usr/sbin/named-xfer
```

The global store for the primary name server hosts file is `/projects/DNS`. Thus, we needed to create this directory and update the automounter configuration, as described in Section 6.2.11, “Tailoring the NFS automounter” on page 235. The NFS exports file for node6 did not need to be updated, because the top level directory /export/projects was already exported.

We changed the /cws/cust/cws01/auto.isp file and added the following line:

```
DNS             node4:/export/projects/DNS
```

To create the directory on node6 and to update all available nodes, we ran the following command sequence:

```
<cws01>#dsh -w node1 mkdir /export/projects/DNS
<cws01>#dsh -w node1 chmod 755 /export/projects/DNS
<cws01>#/cws/cust/common/AMD_entry.ksh
```

The first standard hosts file were created as hosts.isp in the directory /projects/DNS with following entries:

```
#sample ISP hosts file for redbook SG24-6025
192.168.3.1 dns-1.isp.net dns-1
192.168.3.2 dns-2.isp.net dns-2
192.168.3.3 mail-1.isp.net mail-1
192.168.3.4 mail-2.isp.net mail-2
192.168.3.6 ldap-1.isp.net ldap-1
192.168.3.200 www.isp.net www ftp
```

The h2n script execution behavior can be managed through an options file. We created one for each hosts file and placed it into /projects/DNS with a corresponding h2n_options file in the same directory. In this example, the h2n_options.isp file reads the following entries:

```
#sample h2n option file for redbook SG24-6025
-d isp.net
-n 192.168.3
-n 255.253.255
-h dns-1.isp.net
-s dns-1.isp.net
-s dns-2.isp.net
```

```
-m 50:mail-1.isp.net
-m 100:mail-2.isp.net
-u dnsadmin@isp.net
-H /etc/dns/hosts
-v 8
-y
```

We were not able to use the globalized directory /projects to hold the named readable database because the nodes can not write into this NFS automounted directory. On the other hand, de-coupling the services configuration files this way generates a certain level of data protection.

We created a simple shell script, placed it into the /cws/cust/common directory, and named it dns_config.ksh. The script allowed us to update DNS configurations on the dedicated name server nodes. The script copied the hosts file from /projects/DNS into the local directory /etc/dns and converted them, using h2n, into a named readable database files. After conversion, the name server daemon was restarted so that all changes could take effect.

The domain cache file that is usually named db.cache was copied from the globalized definition directory to the local directory. It defines the root name server that propagates the domain name. The content of the file may differ between DNS domains.

The following script allows us to update DNS configuration on the dedicated name server nodes:

```
#!/bin/ksh
#dns_config.ksh script written for redbook SG24-6025
#
JULIEN=$(date +%j)
HOSTNAME=$(hostname)
#
LOCDNS="/etc/dns"
HOSTFILE="${LOCDNS}/hosts"
DNS_PATH="/projects/DNS/"
OPTIONS="${DNS_PATH}/h2n_options"
#
if [ ! -d "${LOCDNS}" ]
then
 mkdir ${LOCDNS}
 cd ${LOCDNS}
else
 cd ${LOCDNS}
 rm hosts
fi
```

```
#
HOSTSLIST=$(ls ${DNS_PATH}/hosts.*)
#
for FILE  in ${HOSTSLIST}
do
echo "processing file ${FILE}"
cp ${FILE} hosts
DOMAIN=$(echo ${FILE} | awk -F.  '{print $2}')
cp ${DNS_PATH}/db.cache.${DOMAIN} db.cache
/cws/cust/common/h2n -f ${OPTIONS}.${DOMAIN}
done
#
/usr/bin/refresh -s named
#
exit
```

At this stage, we were ready to run the script on node1 and populate the first DNS domain, isp.net:

```
<cws01>#dsh -w node1 /cws/cust/common/dns_config.ksh
```

The command started the named server for this session only. But we needed this server to run after a reboot of the system. Thus, we created an entry in the customized startup script rc.include, which is located in the directory /cws/cust/node1/etc/. The corresponding entry in the /etc/inittab file has already been made, as described in Section 6.2.3, “Tailoring firstboot.cust, tuning.cust and script.cust” on page 224.

The /cws/cust/node1/etc/rc.include file shows the new lines:

```
#!/bin/ksh
#node1 rc.include file written for redbook SG24-6025
#
/usr/bin/mknotify -n named -m /cws/cust/common/restart_named.ksh
1 /cws/cust/common/restart_named.ksh
```

1 These lines have been split for redbook printing purposes. In the actual file, they are on a single line.

We then took advantage of the AIX subsystem resource controller (SRC) and the SRC enabled AIX named distribution. For Details on SRC, see the publication *AIX Version 4.3 System Management Concepts: Operating System and Devices* and *AIX Commands Reference, Volume 3,* SC23-4126.

The `mknotify` command adds a notify method definition to the notify object class. The SRC places the name of the unsuccessful subsystem as the first argument to the method and executes the specified method. The method in our example was the script /cws/cust/common/restart_named.ksh:

```
#!/bin/ksh
#restart_named.ksh script written for redbook SG24-6025

DATE=$(date)
SERVICE=named
OPTIONS="-b /etc/dns/named.boot"
HOSTNAME=$(hostname)
LOGPATH="/cws/log/${HOSTNAME}"
#
echo "Restarting ${SERVICE} on ${HOSTNAME} at ${DATE}" >>
${LOGPATH}/${SERVICE}
startsrc -s ${SERVICE} -a "${OPTIONS}"
```

The script restarts the named subsystem and writes a log file entry into the globalized log file directory /cws/log/node1 into the file named. A sample /cws/log/node1/named log file entry look like:

```
Restarting named on node1 at Fri Oct 20 18:35:42 EDT 2000
```

6.3.4 Mail gateway configuration

For implementing the Mail service, we used the standard AIX send mail transport agent (MTA) installed with the AIX fileset bos.net.tcp.client. Mail delivery is based on user related entries with aliases that the database builds using the /etc/aliases local file as data source. For more details on Mail system configuration, refer to *AIX Version 4.3 System Management Guide: Communications and Networks,* SC23-4127.

The global storage for the mail gateway server definition and data source file is /projects/MAIL. Thus, we created this directory and updated the automounter configuration, as already described in Section 6.2.11, “Tailoring the NFS automounter” on page 235.

We changed the /cws/cust/cws01/auto.isp file and added the following line to it:

```
MAIL node4:/export/projects/MAIL
```

To create the directory on node6 and to update all available nodes, we ran the script:

```
<cws01>#dsh -w node4 mkdir /export/projects/MAIL
<cws01>#dsh -w node4 chmod 755 /export/projects/MAIL
<cws01>#/cws/cust/common/AMD_entry.ksh
```

The first standard alias file was created with the name aliases.isp within directory /projects/MAIL, and it read the following entries:

```
dnsadmin: root@node4.sp2.int
ftpadmin: root@node4.sp2.int
webmaster: root@node4.sp2.int
```

The next step is to standardize the sendmail daemon configuration file sendmail.cf. We copied this file into the globalized directory /projects/MAIL:

```
<cws01>#cp /export/projects/MAIL
<cws01>#rcp node1:/etc/sendmail.cf sendmail.cf
<cws01>#cp sendmail.cf sendmail.cf.org
```

We removed the comments from the following lines within the /projects/MAIL/sendmail.cf file:

```
Fw-o /etc/mail/sendmail.cw
FR-o /etc/mail/relay-domains
```

and changed the line O AliasFile=/etc/aliases to read:

```
O AliasFile=/etc/mail/aliases
```

The sendmail.cw and relay-domains files were created from scratch within the directory /projects/MAIL. The following initial setup entries were added for the sample implementation:

- sendmail.cw

  ```
  www.isp.net
  www-1.isp.net
  www-2.isp.net
  dns-1.isp.net
  dns-2.isp.net
  mail-1.isp.net
  mail-2.isp.net
  ldap-1.isp.net
  ldap-2.isp.net
  ```

- relay-domains

```
isp.net
sp2.int
```

The same limitations apply, as described in Section 6.3.3, “Domain name system configuration” on page 245, for writing application readable configuration files into the globalized directory /projects/MAIL. Thus, we needed a simple shell script placed into the /cws/cust/common directory named mail_config.ksh to update sendmail configurations on the dedicated mail gateway nodes. The script copied the aliases file from /projects/MAIL into the local directory /etc/mail and converted them using the command `sendmail -bi` into the sendmail readable database files. After conversion, the sendmail daemon was restarted so that all changes would be in effect.

The following script updates send mail configuration on the dedicated gateway nodes:

```
#!/bin/ksh
#mail_config.ksh script written for redbook SG24-6025
#
JULIEN=$(date +%j)
HOSTNAME=$(hostname)
#
LOCMAIL="/etc/mail"
ALIASES="${LOCMAIL}/aliases"
MAIL_PATH="/projects/MAIL"
#
if [ ! -d "${LOCMAIL}" ]
then
 mkdir ${LOCMAIL}
 cd ${LOCMAIL}
else
 cd ${LOCMAIL}
 rm aliases
fi
#
cp ${MAIL_PATH}/sendmail.cf /etc/sendmail.cf
#
ALIASESLIST=$(ls ${MAIL_PATH}/aliases.*)
#
for FILE  in ${ALIASESLIST}
do
echo "processing file ${FILE}"
cat /etc/aliases > ${ALIASES}
cat ${FILE} >> ${ALIASES}
done
```

```
#
/usr/sbin/sendmail -bi
refresh -s sendmail
#
exit
```

At this stage, we ran the script on node1 and restarted the mail gateway server:

```
<cws01>#dsh -w node1 /cws/cust/common/mail_config.ksh
```

The command started the mail gateway server only for this session, but we needed this server to run after a reboot of the system as well. Thus, we needed to create an entry within the customized startup script rc.include, which is located in the directory /cws/cust/node1/etc/ for node1. The corresponding entry in the /etc/inittab file has already been created, as described in Section 6.2.3, “Tailoring firstboot.cust, tuning.cust and script.cust” on page 224.

By default, the sendmail daemon is started during the initial run level 2 execution of /etc/rc.tcpip. But as described within Section 6.2.3, “Tailoring firstboot.cust, tuning.cust and script.cust” on page 224 and Section 6.2.9, “Basic security setup” on page 232 the default rc.tcpip file was overwritten by our version. See Appendix B.1.1, “rc.tcpip file” on page 287 for details on the customized rc.tcpip file.

We added the following lines to the /cws/cust/node1/etc/rc.include file so that sendmail was started during the boot time of the node:

```
/usr/bin/mknotify -n sendmail -m /cws/cust/common/restart_sendmail.ksh
1  /cws/cust/common/restart_sendmail.ksh
```

1 These lines have been split for redbook printing purposes. In the actual file, they are on a single line.

With the script restart_sendmail.ksh, we took advantage of AIX SRC, as described in Section 6.3.3, “Domain name system configuration” on page 245:

```
#!/bin/ksh
#restart_sendmail.ksh script written for redbook SG24-6025
DATE=$(date)
SERVICE=sendmail
OPTIONS="-bd -q=5m"
HOSTNAME=$(hostname)
```

```
LOGPATH="/cws/log/${HOSTNAME}"
#
echo "Restarting ${SERVICE} on ${HOSTNAME} at ${DATE}"
1 >> ${LOGPATH}/${SERVICE}
startsrc -s ${SERVICE} -a "${OPTIONS}"
```

1 These lines have been split for redbook printing purposes. In the actual file, they are on a single line.

The script restarts the named subsystem and writes a log file entry into the globalized log file directory /cws/log/node1 into file named.

A sample /cws/log/node1/sendmail log file entry looks like:

```
Restarting sendmail on node1 at Fri Oct 20 18:35:42 EDT 2000
```

6.3.5 FTP configuration

In order to use the ftp configuration, we need a ftp service. We used the *WU-FTP* server, as described in Section 6.2.4, “Creating the common tools data repository on the CWS” on page 227, because this free available FTP server allowed us to configure and use:

- Free welcome and directory messages
- Anonymous ftp connections
- Control directory access for upload and download streams
- Group based password protected access

For details on WU-FTP, see `http://www.wu-ftpd.org/`

The global store for the ftp server configuration files is /projects/FTP. Thus, we created this directory and updated the automounter configuration, as described in Section 6.2.11, “Tailoring the NFS automounter” on page 235. The NFS exports file for node6 did not need to be updated, because the top level directory /export/projects was already exported.

We changed the /cws/cust/cws01/auto.isp file and added the line:

```
FTP node4:/export/projects/FTP
```

To create the directory on node6 and update all available nodes, we ran the following command sequence on the CWS:

```
<cws01>#dsh -w node4 mkdir /export/projects/FTP
<cws01>#dsh -w node4 chmod 755 /export/projects/FTP
<cws01>#/cws/cust/common/AMD_entry.ksh
```

The main ftp configuration file is called ftpaccess. We copied this file from the default distribution directory /usr/local/lib/wu-ftpd-2.6.1/examples into the general ftp configuration directory /projects/FTP/ as ftpaccess.node3:

```
<cws01>#cp /usr/local/lib/wu-ftpd-2.6.1/examples
1  /projects/FTP/ftpaccess.node3
```

1 These lines have been split for redbook printing purposes. In the actual file, they are on a single line.

The /projects/FTP/ftpaccess.node3 file was changed to read the following basic configuration statements:

```
#sample wu-ftp ftpaccess file for redbook SG24-6025
#
class   all   real,guest,anonymous  *
limit   all   10   Any              /projects/FTP/msg.dead
readme  README*    login
readme  README*    cwd=*
banner  /projects/FTP/banner
message /welcome.msg            login
message .message                cwd=*
compress        yes             all
tar             yes             all
log commands real
log transfers anonymous,real inbound,outbound
shutdown /projects/FTP/shutmsg
email ftpadmin@isp.net
```

The banner file holds the general message displayed on the ftp client login panel. Thus, we had only one general banner file for all possible ftp servers on our site, with the following initial content:

```
******************************************************
*           welcome                                  *
******************************************************
Questions ? : %E
```

Note

For details on configuring WU-FTP, see the manual pages and the FAQ pages on http://www.wu-ftpd.org/.

The standard AIX ftp daemon is, by default, started through the inetd daemon entry within the inetd.conf configuration file. This entry does not comply to the one required by WU-FTP. Thus, we needed to change the line within the nodes globalized configuration directory /cws/cust/node3/etc/inetd.conf to read:

```
ftp     stream  tcp    nowait  root    /usr/local/bin/ftpd        ftpd  -l -a -i -o -d -L
```

To update the inetd daemon, we used a simple script which we created in the generalized directory /cws/cust/common called refresh_inetd.ksh. The script copies the node dependant inetd.conf configuration file from /cws/cust/node3/etc/ to the local /etc/inetd.conf file and refreshes the inetd SRC Subsystem:

```
#!/bin/ksh
#refresh_inetd script written for redbook SG24-6025
DATE=$(date)
SERVICE=inet
OPTIONS=""
HOSTNAME=$(hostname)
LOGPATH="/cws/log/${HOSTNAME}"
INETDP="/cws/cust/${HOSTNAME}"
INETD="/etc/inetd.conf"
#
cp ${INETDP}${INETD} ${INETD}
#
echo "refreshing ${SERVICE} on ${HOSTNAME} at ${DATE}" >> ${LOGPATH}/${SERVICE}
if [ ! -z "${OPTIONS}" ]
then
 refresh -s ${SERVICE} -a "${OPTIONS}"
else
 refresh -s ${SERVICE}
fi
```

1 These lines have been split for redbook printing purposes. In the actual file, they are on a single line.

For the changes to take effect, we executed the script in node3:

```
<cws01>#dsh -w node3 /cws/cust/common/refresh_inetd.ksh
```

The compiled WU-FTP distribution should have the configuration files placed into the /etc directory on the local node. Thus, we created a script called

ftp_config.ksh in the global directory /cws/cust/common. This scripts copies the nodes ftp configuration files out of the generalized ftp configuration directory /projects/FTP into the local node /etc directory. Furthermore, the script check if the globalized ftp log file is available:

```
#!/bin/ksh
#ftp_config.ksh script written for redbook SG24-6025
#
JULIEN=$(date +%j)
HOSTNAME=$(hostname)
#
LOCFTP="/etc"
FTP_PATH="/projects/FTP"
LOGP="/cws/log/${HOSTNAME}"
XFERL="${LOGP}/xferlog"
CHKL="${LOGP}/ckconfig.out"
LOGL="/var/adm/xferlog"
#
if [ -f ${XFERL} ]
then
 if [ ! -a ${LOGL} ]
 then
  ln -sf ${XFERL}  ${LOGL}
 fi
else
 touch ${XFERL}
 ln -sf ${XFERL}  ${LOGL}
fi
#
for FILE  in ftpaccess
do
echo "processing file ${FILE}"
cp ${FTP_PATH}/${FILE}.${HOSTNAME} /etc/${FILE}
chmod 644 /etc/${FILE}
done
#
/usr/local/bin/ckconfig  > ${CHKL}
#
exit
```

To finally activate all the changes, we ran the script on node3:

```
<cws01>#dsh -w node8 /cws/cust/common/ftp_config.ksh
```

Basic verification tests can be performed using the AIX `ftp` command to login into node3. The final output is shown in the following sample output:

```
<cws01># ftp node3
Connected to node3.
220-*********************************************************
220-*           welcome                                     *
220-*********************************************************
220-Questions ? : ftpadmin@isp.net
220-
220 node3 FTP server (Version wu-2.6.1(1) Thu Jul 20 19:10:14
1  DFT 2000) ready.
Name (node3:root): ftp
331 Guest login ok, send your complete e-mail address as password.
Password:
230 Guest login ok, access restrictions apply.
ftp>T
```

1 These lines have been split for redbook printing purposes. In the command and output sequence, they are on a single line.

6.4 Application deployment

After installing and configuring the basic AIX installation and basic services such as NFS, mail, and ftp, we were ready for application deployment within the RS/6000 SP environment.

6.4.1 Installing IBM Network Dispatcher

The IBM Network Dispatcher (eND) Code has been downloaded into /cws/install/Wedge as Part of the WebSphere(TM) EveryPlace Server for Multiplatforms IBM Product.

Node1 has been defined as the Primary eND Server and node2 as the backup eND Server. Thus, we installed the eND code on both servers simultaneously by taking advantage of the working collective file, as described in Section 6.2.12, “Update the working collective file WCOLL” on page 237.

To do the installation only on node1 and node2, we commented all the lines in the $HOME/WCOLL file except for node1 and node2. This allowed us to use the following command sequence:

```
<cws01>#dsh -i /cws/install/WEdge/setup -ndinstall
1 -default -installonly
```

1 These lines have been split for redbook printing purposes. In the command sequence, they are on a single line.

The command output looks like:

```
node2: Running ND Installation...
node2: Details on the install are located in the log file:
1 /usr/lpp/nd/install.log
node2: ND Installation complete.
node1: Running ND Installation...
node1: Details on the install are located in the log file:
1 /usr/lpp/nd/install.log
node1: ND Installation complete.
```

1 These lines have been split for redbook printing purposes. In the console output, they are on a single line.

The ND installation log file can be examined using:

```
<cws01>#dsh -w node1 pg  /usr/lpp/nd/install.log
```

The install log was copied into the globalized log directory for later archival purposes:

```
<cws01>#dsh -w node1 cp /usr/lpp/nd/install.log
1 /cws/log/node1/install.nd.log
<cws01>#dsh -w node2 cp /usr/lpp/nd/install.log
1 /cws/log/node2/install.nd.log
```

1 These lines have been split for redbook printing purposes. In the actual file, they are on a single line.

The eND configuration files for operating a node as an eND server were created in the directory /cws/cust/node1/nd. Table 6 shows the configuration files and their general task within the eND environment.

Table 6. Network Dispatcher configuration scripts

Script Name	Script Task
start.cfg	Executed to start eND.
goActive	Executed by eND when eND goes in Active state.
goStandby	Executed by eND when eND goes in Standby state.
goInOp	Executed by eND when eND is stopped.

We copied the sample eND configuration scripts located in the directory /usr/lpp/nd/dispatcher/samples from node1 into the globalized ND node1 configuration tree /cws/cust/node1/nd:

```
<cws01>#cd /cws/cust/node1/nd
<cws01>#dsh -w node1 rcp /usr/lpp/nd/dispatcher/samples/
1  goActive.samplegoActive
<cws01>#dsh -w node1 rcp /usr/lpp/nd/dispatcher/samples/
1  goStandBy.samplegoStandBy
<cws01>#dsh -w node1 rcp /usr/lpp/nd/dispatcher/samples/
1  goInOp.samplegoInOp
<cws01>#dsh -w node1 rcp /usr/lpp/nd/dispatcher/samples/
1  configuration.sample start.cfg
```

1 These lines have been split for redbook printing purposes. In the actual command sequence, they are on a single line.

We changed the files to adopt the settings defined in the architectural layout. The final version of all the scripts listed in Table 6 can be found in Appendix B.3, “Network Dispatcher configuration files and sample listing” on page 297. The start.cfg script for the primary eND node looks like:

```
#!/bin/ksh
#Primary nd startup script written for redbook SG24-6025
ndserver start
ndcontrol executor start
NFA=192.168.3.1
CLUSTER=192.168.3.200
ndcontrol executor set nfa $NFA
```

```
ndcontrol cluster add $CLUSTER
ndcontrol port add $CLUSTER:20+21+80
SERVER1=192.168.3.3
SERVER2=192.168.3.4
ndcontrol server add $CLUSTER:20+21+80:$SERVER1+$SERVER2
ndcontrol manager start
ndcontrol advisor start ftp 21
ndcontrol advisor start http 80
ndcontrol manager proportions 58 40 2 0
#
ndcontrol set loglevel 5
ndcontrol set logsize unlimited
ndcontrol manager loglevel 5
#
ndcontrol highavailability heartbeat add 192.168.3.1 192.168.3.2
ndcontrol highavailability reach add 192.168.3.130
ndcontrol highavailability backup add backup primary 11001
```

We created a simple script called script ND_update_server.ksh in the directory /cws/cust/common, which copies the script shown in Table 6 into the local node directory /usr/lpp/nd/dispatcher/bin. After the scripts had been copied, the nd startup script start.cfg was automatically executed:

```
#!/bin/ksh
#ND_update_server.ksh script written for redbook SG24-6025

HOST=$(hostname)
#
LOCND="/usr/lpp/nd/dispatcher/bin/"
PRIM="node1"
BACK="node2"
#
cd ${LOCND}
cp /cws/cust/${HOST}/nd/* .
chmod 755 go* start.cfg
#
if [ "${HOST}" = "${PRIM}" ]
then
${LOCND}/start.cfg
elif [ "${HOST}" = "${BACK}" ]
then
${LOCND}/start.cfg
else
echo "noop not configured as ND Server"
fi
#
exit
```

To start up eND and check its status, we executed the commands listed on node1:

```
<cws01>#dsh -w node1 /cws/cust/common/ND_update_server.ksh |
1  tee -a /cws/logs/node1/ND.startup &
<cws01>#dsh -w node1 ndcontrol manager  report
```

1 These lines have been split for redbook printing purposes. In the actual command sequence, they are on a single line.

The sample output of the command `dsh -w node1 ndcontrol manager report` is listed in Appendix B.3.5, “Sample output from eND command ndcontrol manager report” on page 299. The command started the eND server only for this session, but we need this server to run after a reboot of the system as well. Thus, we needed to create an entry within the customized startup script rc.include, which is located in the directory /cws/cust/node1/etc/. The corresponding entry within the /etc/inittab file has already been made, as described in Section 6.2.3, “Tailoring firstboot.cust, tuning.cust and script.cust” on page 224.

The /cws/cust/node1/etc/rc.include file shows the newly created lines:

```
/usr/lpp/nd/dispatcher/bin/start.cfg > /cws/logs/node1/ND.startup
```

To set the node2 function as an eND backup server in advance, we created the same entry within the node2 startup file /cws/cust/node2/etc/rc.include.

Adding the backup eND server on node2 required the copying of the eND configuration files from the globalized configuration directory in node1 /cws/cust/node1/nd into the /cws/cust/node2/nd eND configuration directory in node 2:

```
<cws01>#cd /cws/cust
<cws01>#cp -R node1 node2
<cws01>#cd node2/nd
```

We changed the eND high availability lines in the start.cfg file in node2 eND configuration directory /cws/cust/node2/nd to read:

```
ndcontrol highavailability heartbeat add 192.168.3.5 192.168.3.200
ndcontrol highavailability backup add backup manual 11001
```

For details on the ND configuration scripts, see Appendix B.3.1, “start.cfg file for backup eND server” on page 297 and Appendix B.3.2, “goActive file” on

page 297. At this stage, we were able to start up the backup eND Server and check its functionality:

```
<cws01>#dsh -w node2 /cws/cust/common/ND_update_server.ksh |
1 tee -a /cws/logs/node2/ND.startup &
<cws01>#dsh -w node2 ndcontrol highavailability status
High Availability Status:
-------------------------
Role ................. Backup
Recovery strategy .... Manual
State ................ Standby
Sub-state ............ Synchronized
Port ................. 11001
Preferred target ..... 192.168.3.130

Heartbeat Status:
-----------------
Count ................ 1
Source/destination ... 192.168.3.2/192.168.3.1

Reachability Status:
--------------------
Count ................ 1
Address .............. 192.168.3.200
```

1 These Lines have been split for redbook printing purposes. In the actual command sequence, they are on a single line.

6.4.2 Installing IBM HTTP Server

The IBM HTTP Server (IHS) code has been downloaded into the /cws/install/IHS directory as part of WebSphere(TM) EveryPlace Server for Multiplatforms IBM Product. Node3 and node4 were defined as IHS HTTP servers. Thus, we installed IHS code on both servers simultaneously by taking advantage of the working collective file, as described with Section 6.2.12, “Update the working collective file WCOLL” on page 237.

To install the software, we commented all the lines in the $HOME/WCOLL file except for the ones with node3 and node4. This allowed us to use the following command sequence:

```
<cws01>#dsh -w i /usr/sbin/installp -acQ -g -d
1 /cws/install/IHS  http_server.admin http_server.base.rte
1 http_server.frca http_server.ssl.56
1 http_server.msg.en_US.ssl.core http_server.msg.en_US.admin
1 http_server.modules.ldap -X -e /tmp/IHS.log
```

1 These lines have been split for redbook printing purposes. In the actual command sequence, they are on a single line.

A more comfortable way to initiate the batch installation is to use a shell script, such as:

```
#!/bin/ksh
#IHS_install.ksh script written for redbook SG24-6025
HOSTNAME=$(hostname)
#
/usr/sbin/installp -acQ -g -d /cws/install/IHS \
http_server.admin http_server.base.rte http_server.frca \
http_server.ssl.56 http_server.msg.en_US.ssl.core \
http_server.msg.en_US.admin \
-X -e /cws/log/${HOSTNAME}/IHS.install
#
exit 0
```

For the above given task, the command sequence to install and control the installation from the CWS looks like:

```
<cws01>#dsh -i /cws/cust/common/IHS_install.ksh &
<cws01>#tail -f /cws/log/node3/IHS.install
```

To further generalize the configuration of the other IHS server, we copied the configuration files from the default HTTP server configuration directory /usr/HTTPServ/conf into /projects/www/IHSConf:

```
<cws01># rcp node3:/usr/HTTPServer/conf/*.conf
1 /projects/www/IHSConf
```

1 These lines have been split for redbook printing purposes. In the actual command sequence, they are on a single line.

We changed the ServerAdmin, Directory, and DocumentRoot HTTP Server configuration directive within the HTTP Server configuration file /projects/www/IHSConf/httpd.conf to read the server www.isp.net (IP-Address 192.168.3.200).

```
ServerAdmin webmaster@www.isp.net
<Directory /projects/www/html>
DocumentRoot /projects/www/html
```

We added the ServerName directive to the /projects/www/IHSCon/httpd.conf file to read:

```
ServerName 192.186.3.200
```

We used the script /cws/cust/common/IHS_www_update_server.ksh to update and restart the HTTP server on the HTTP server nodes.The script sets the IP aliases for the HTTP server www.isp.net and copies the HTTP server configuration files out of the globalized directory into the local nodes HTTP server default directory. The configuration is verified and the HTTP server is restarted. The script is as follows:

```
#!/bin/ksh
HOST=$(hostname)
LOCIHS="/usr/HTTPServer/conf"
BINIHS="/usr/HTTPServer/bin"
#
/etc/ifconfig lo0 alias 192.168.3.200 netmask 255.255.255.0
#
cd  ${LOCIHS}
cp /projects/www/IHSConf/* .
#
${BINIHS}/adminctl configtest
if [ $? -ne 0 ]
then
echo "adminctl configtest returned with $?"
else
${BINIHS}/adminctl restart
fi
#
${BINIHS}/apachectl configtest
if [ $? -ne 0 ]
then
echo "apachectl configtest returned with $?"
else
${BINIHS}/apachectl restart
fi
#
exit
```

We used the script on the CWS to start the server on node3 and node4:

```
<cws01>#dsh -w node3 /cws/cust/common/IHS_www_update_server.ksh
<cws01>#dsh -w node4 /cws/cust/common/IHS_www_update_server.ksh
```

The output of the command displayed is:

```
node3: /usr/HTTPServer/bin/adminctl restart: admin http
1 not running, trying to start
node3: /usr/HTTPServer/bin/adminctl restart: admin http
1 started
node3: /usr/HTTPServer/bin/apachectl restart: httpd not
1 running, trying to start
node3: /usr/HTTPServer/bin/apachectl restart: httpd started
node3: Syntax OK
node3: Syntax OK
```

1 These lines have been split for redbook printing purposes. In the actual command and output sequence, they are on a single line.

The command started the IHS server only for this session, but we need this server to run after a reboot of the system too. In addition, we need to alias the IP Address for the server www.isp.ne onto the network interface on the local node connected to the external network. Thus, we created an entry within the customize startup script rc.include, which is located in the directory /cws/cust/node3/etc/ and /cws/cust/node4/etc/. The corresponding entry in the /etc/inittab file has already been made, as described with Section 6.2.3, “Tailoring firstboot.cust, tuning.cust and script.cust” on page 224.

The /cws/cust/node3/etc/rc.include and /cws/cust/node4/etc/rc.include files show the added lines:

```
/etc/ifconfig lo0 alias 192.168.3.200 netmask 255.255.255.0
/usr/HTTPServer/bin/httpd
```

As soon as HTML content has been placed into the globalized directory /projects/www/html, the HTTP Server serves static HTML pages for the server www.isp.net under the URL `http://www.isp.net`.

The following sample shows a simple HTML file that we accessed through Web browser, as shown in Figure 70 on page 266:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4//EN">
<!-- -->
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
1 charset=iso-8859-1">
<title>Redbook SG24-6025 Sample Webpage</title>
</head>
<body>
<H1>Hello World</H1>
```

```
</body>
<a href="mailto:webmaster@isp.net">Feedback ?</a>
</html>
```

1 These lines have been split for redbook printing purposes. In the actual file, they are on a single line.

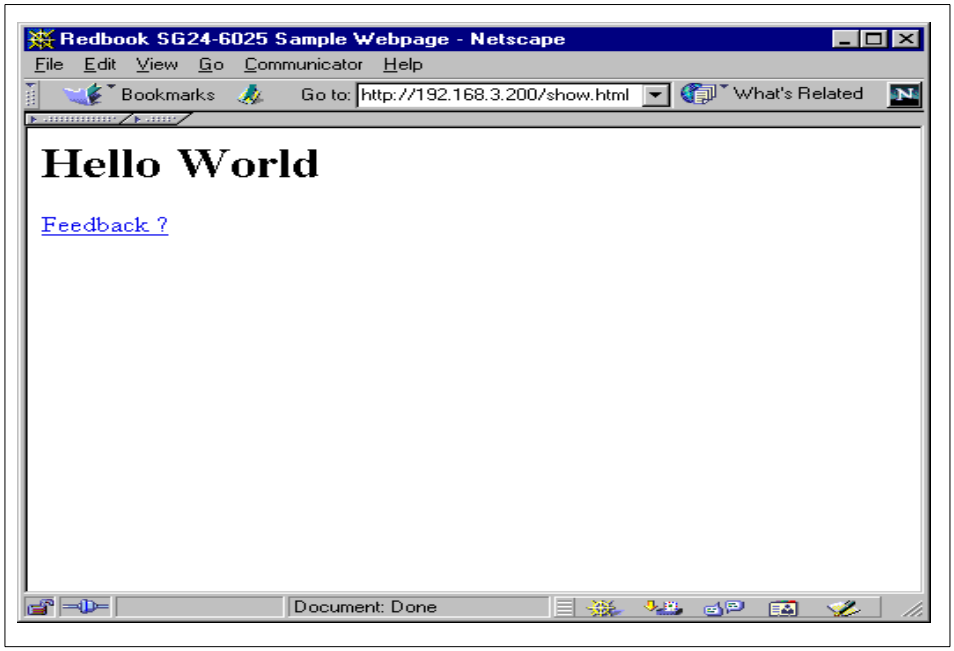


Figure 70. Web browser access to sample the Web site www.isp.net

6.4.3 Installing DB2 V71 Extended Enterprise Edition

The IBM DB2 V71 code has been downloaded in /cws/install/DB271 as part of the WebSphere(TM) EveryPlace Server for Multiplatforms IBM Product. We loaded the DB2 FixPak 1 into the directory /cws/install/DB271/FP1 from the IBM support site `http://www-4.ibm.com/software/data/db2/library/`.

For unattended installation purposes, DB2 delivers the installation configuration file called the response file. Sample files are delivered with the code distribution for later use. We copied the default delivered file db2udbeee.rsp into the globalized directory /cws/install/DB271 and changed it to adopt our needs:

```
<cws01>#cd /cws/install/DB271/
<cws01>#cp ./db2/install/samples/db2udbeee.rsp db2udbeee.rsp
```

See the *DB2 UDB Installation and Configuration Supplement V6*, GC09-2857, for details on the response file syntax.

We changed the DB2 installation response file to install a DB2 server instance with the Java and replication enabled support features:

```
*
* Changed for redbook SG24-6025
*
* Product Installation
* --------------------
INSTALL_SOURCE_PATH       = /cws/install/DB271/db2
PROD                      = UDB_EEE
COMP                      = JAVA_SUPPORT
COMP                      = DIRECTORY_ACCESS_PROTOCOL
COMP                      = REPLICATION
* Instance Creation Settings
* --------------------------
DB2.USER_NAME                          = db2inst1
DB2.GROUP_NAME                         = www
DB2.HOME_DIRECTORY                     = /home/db2inst1
DB2.AUTHENTICATION                     = SERVER
DB2.SAMPLE                             = NO
DB2.AUTOSTART                          = YES
DB2.SVCENAME                           = db2cdb2inst1
DB2.PORT_NUMBER                        = 50000
DB2.FCM_PORT_NUMBER                    = 60000
* Fenced User Creation Settings
* -----------------------------
UDF.USER_NAME                          = db2inst1
UDF.GROUP_NAME                         = www
UDF.HOME_DIRECTORY                     = /home/db2inst1
* --------------------------------
DB2COMM                                 = TCPIP
```

We need to take care that the User IDs, as defined in Section 6.2.10, “File collections for /etc/passwd and /etc/group updates” on page 233, are used in this file.

We placed a script named DB2_install.ksh in the directory /cws/install/DB271, which creates the needed local file systems for storing DB2 and performs the

unattended installation of the product itself. The initial setup definition of the system on node5 requires only one of the SSA disks to be used for holding the volume group named db2_vg. Thus, we went to the first available hard disk on the system named hdisk3:

```
#!/bin/ksh
#DB2_install.ksh created for redbook SG24-6025
HOSTNAME=$(hostname)
#Creating db2_vg with PPSize=32MB
mkvg -f -y'db2_vg' -s'32' hdisk3
#
#Creating the DB2 Logocal Volumes
mklv -y'db2_udb71_lv' db2_vg 5
#Creating the Associated Filesystems
crfs -v jfs -d db2_udb71_lv  -m /usr/lpp/db2_07_01 \
-a frag='4096' -a nbpi='4096' -A'yes' \
-p'rw' -t'no' 1>/dev/null 2>/dev/null
chown root.system /usr/lpp/db2_07_01
chmod 775 /usr/lpp/db2_07_01
mount /usr/lpp/db2_07_01
chown root.system /usr/lpp/db2_07_01
chmod 775 /usr/lpp/db2_07_01
#for DB2 HomeDirs
mklv -y'db2_home_lv' db2_vg  1
crfs -v jfs -d db2_home_lv  -m /home/db2inst1 \
-a frag='4096' -a nbpi='4096' -A'yes' -p'rw' \
-t'no' 1>/dev/null 2>/dev/null
mount /home/db2inst1
chown db2inst1.www /home/db2inst1
chmod 755 /home/db2inst1
/cws/install/DB271/db2setup -r /cws/install/DB271/db2udbeee.rsp
cp /tmp/db2*.log /cws/log/${HOSTNAME}
```

The installation was performed on the node using the command:

```
<cws01>#dsh -w node6 /cws/install/DB271/DB2_install.ksh
```

The installation log file can be viewed using:

```
<cws01>#dsh -w node6 pg /tmp/db2setup.log
```

We then upgraded DB2 to Fixpak level1 from the directory /cws/install/DB271/FP1:

```
<cws01>#dsh -w node6 /usr/lib/instl/sm_inst installp_cmd
1 -a -d /cws/install/DB271/FP1 -f _update_all -g -X
1 -e /cws/log/node5/DB2FP1.log
```

1 These lines have been split for redbook printing purposes. In the actual command sequence, they are on a single line.

The last step was to ensure that the DB2 shared libraries were available, in general, on node6:

```
<cws01>dsh -w node6 /usr/lpp/db2_07_01/cfg/dn2ln
```

At this point, we need the DB2 client connection over IP to enable applications, which require DB2 databases, to communicate with the DB2 server on node6. We defined a service name and a TCP/IP port. This was achieved by adding the following sample statement to the nodes local /etc/services file:

```
DB2_ldapdb2     60002/tcp   # Connection port for DB2 LDAP instance
```

We did this, in advance, by placing the generalized services file into the directory /cws/cust/node6/etc before installing the node. During execution of the script firstbot.cust, the cws/cust/node6/etc/service file was copied to /etc/services. Please see Appendix B.1.3, “Services file” on page 291 for all DB2 customized entries within this file.

6.4.4 LDAP installation and configuration

The IBM SecureWay Directory Server code has been downloaded into /cws/install/LDAP32 as part of the WebSphere(TM) EveryPlace Server for Multiplatforms IBM Product distribution. DB2 V71 and IHS have been installed using the methods described in Section 6.4.2, “Installing IBM HTTP Server” on page 262 and Section 6.4.3, “Installing DB2 V71 Extended Enterprise Edition” on page 266.

The IBM LDAP server, by default, requires a new DB2 instance installed named ldapdb2, which we arranged to reside in the user’s home directory /home/ldapdb2. We placed this directory on its own logical volume, created on the volume group named db2_vg.

The installation and post installation of LDAP was done using a shell script located with /cws/cust/common named LDAP_install.ksh:

```
#LDAP_install.ksh created for redbook SG24-6025
HOSTNAME=$(hostname)
INST="/cws/install/LDAP32"
#
mklv -y'ldapdb2_home_lv' rootvg  10
crfs -v jfs -d ldapdb2_home_lv  -m /home/ldapdb2 -a frag='4096' -a nbpi='4096' -A'yes'
-p'rw' -t'no'
mount /home/ldapdb2
chown ldapdb2.www /home/ldapdb2
chmod 755 /home/ldapdb2
touch /home/ldapdb2/.profile
chmod 755 /home/ldapdb2/.profile
#
/usr/sbin/installp -acQX -g -e /cws/log/${HOSTNAME}/LDAP.install.log -d ${INST}
ldap.server.rte ldap.server.admin ldap.html.en_US
.config ldap.client
#
ldapcfg  -l /home/ldapdb2 1> /cws/log/${HOSTNAME}/LDAPCFG.log
2>/cws/log/${HOSTNAME}/LDAPCFG.log
ldapcfg -s ibmhttp -f /usr/HTTPServer/conf/httpd.conf 1> /cws/log/${HOSTNAME}/LDAPCFG.log
2>/cws/log/${HOSTNAME}/LDAPCFG.log
ldapcfg  -u"cn=root" -psecret 1> /cws/log/${HOSTNAME}/LDAPCFG.log
2>/cws/log/${HOSTNAME}/LDAPCFG.log
#
ldapcfg -w ldapdb2
#
cp /tmp/ldacfg.out /cws/log/${HOSTNAME}/ldacfg.out
#
su - ldapdb2 -c "db2iauto -on ldapdb2"
su - ldapdb2 -c "db2 "update dbm CONFIGURATION USING SVCENAME DB2_ldapdb2""
su - ldapdb2 -c "db2stop"
su - ldapdb2 -c "db2start"
#
ln -sf /usr/bin/java  /usr/ldap/web/cgi-bin/java
#
/usr/bin/slapd 1> /cws/log/${HOSTNAME}/slapd.log 2>/cws/log/${HOSTNAME}/slapd.log
#
/usr/HTTPServer/bin/apachectl restart
#
exit
#
```

The IHS configuration updates need only to be done once, because we shared the httpd configuration file, as described in Section 6.4.2, “Installing IBM HTTP Server” on page 262. The following command executes LDAP_install.ksh:

```
<cws01>#dsh -w node6 /cws/cust/common/LDAP_install.ksh |
1  tee -a /cws/log/node/LDAP_install.out
<cws01>tail -f /cws/log/node/LDAP_install.out
```

1 These lines have been split for redbook printing purposes. In the actual command sequence, they are on a single line.

Before going further, we checked the log file entries and the connectivity to the DB2 database on the node:

```
<cws01>#dsh -w node6 pg /tmp/slapd.errors
10/27/00 12:06:50 Configuration read securePort 636.
10/27/00 12:07:08 Plugin of type EXTENDEDOP is successfully
1loaded from libevent.a.
10/27/00 12:07:08 Plugin of type EXTENDEDOP is successfully
1loaded from libtranext.a.
10/27/00 12:07:08 Plugin of type PREOPERATION is successfully
1loaded from libDSP.a.
10/27/00 12:07:08 Plugin of type EXTENDEDOP is successfully
1loaded from libevent.a.
10/27/00 12:07:08 Plugin of type AUDIT is successfully loaded from
/lib/libldapaudit.a.
10/27/00 12:07:10 Plugin of type EXTENDEDOP is successfully
1loaded from libevent.a.
10/27/00 12:07:10 Plugin of type DATABASE is successfully
1loaded from /lib/libback-rdbm.a.
10/27/00 12:07:54 Non-SSL port initialized to 389.
10/27/00 12:07:54 Local UNIX socket name initialized to
1/tmp/s.slapd.
10/27/00 12:08:04 SecureWay Directory, Version 3.2
        slapd started.
<cws01>#
```

1 These Lines have been split for redbook printing purposes. In the actual command sequence, they are on a single line.

The following command checks the connectivity of the DB2 database:

```
<cws01>#dsh -w node6 su - ldapdb2 -c "db2 "connect to ldapdb2""

   Database Connection Information

 Database server        = DB2/6000 7.1.0
 SQL authorization ID   = LDAPDB2
 Local database alias   = LDAPDB2

<cws01>#
```

The configuration currently does not provide an option for starting the LDAP server at system boot time. However, this can be achieved by manually adding a line to local /etc/inittab file:

```
ldapd:2:once: /bin/slapd > /dev/console 2>&
```

Since we used globalized startup files within an entry in the node's /etc/inittab file we, added following line to the /cws/cust/node/etc/rc.include:

```
#!/bin/ksh
#node rc.include written for redbook SG24-6025
/bin/slapd > /cws/log/${HOSTNAME}/slapd.out
```

We tested the function using the URL `http:ldap-1.isp.net/ldap/index.html`,and the result is shown in Figure 71.

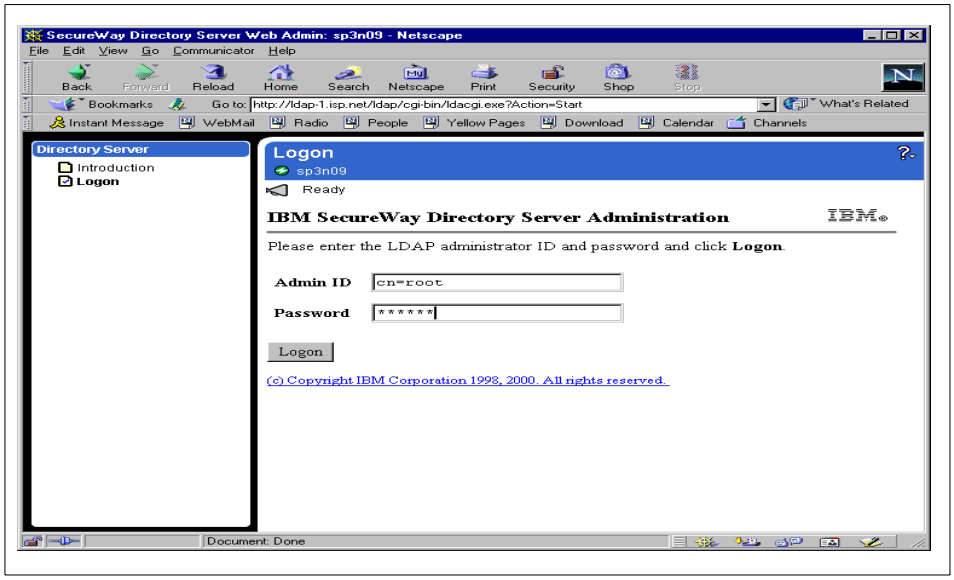


Figure 71. Sample LDAP server Web browser login panel

Figure 72 on page 273 shows the final Web page after basic authentication.

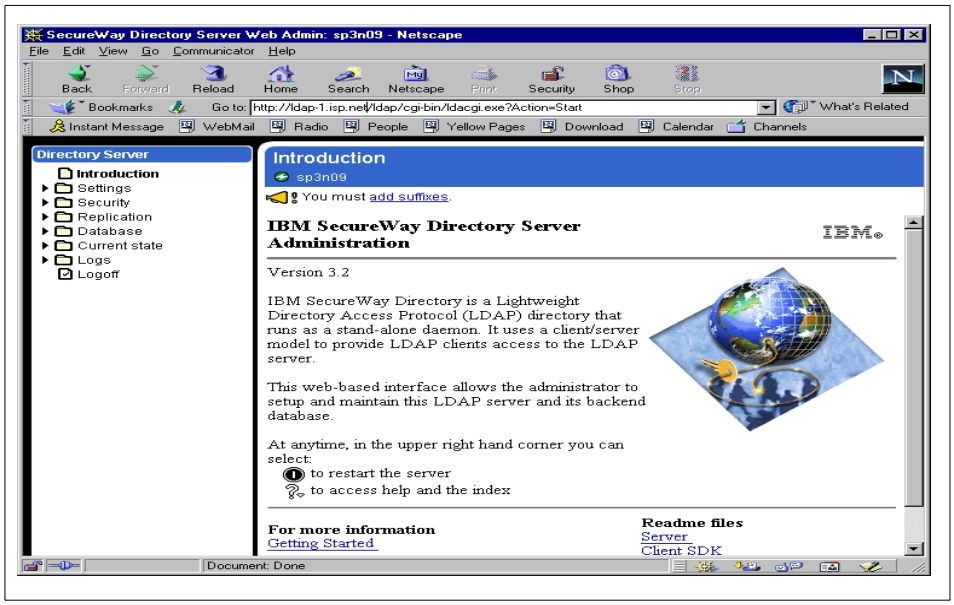


Figure 72. Sample LDAP server administrative page

The LDAP configuration file sldap32.conf that is located in the /etc directory on the local file system of node6 will be copied to the globalized configuration tree /cws/cust/node6/etc for later customization:

```
<cws01>#cd /cws/cust/node6/etc
<cws01>#rcp node6:/etc/slapd32.conf slapd32.conf
```

We added the LDAP directory suffixes within the LDAP server configuration file in the globalized directory /cws/cust/node9/etc/slapd32.conf. Therefore, we added the file below the line that reads ibm-slapdSuffix: cn=localhost with a new line that reads:

```
ibm-slapdSuffix: o=BORG, c=DE
```

We updated the sldapd configuration file on the node running the master LDAP server. A refresh of the server is not needed after updating the configuration file:

```
<cws01>#cd /cws/cust/node6/etc
<cws01>#dsh -w node6 cp etc/slapd32.conf etc/slapd32.conf.old
<cws01>#rcp slapd32.conf node6:/etc/slapd32.conf
```

We checked the correct entry using the URL:

`http://ldap-1.isp.net/ldap/cgi-bin/ldacgi.exe`

as shown in Figure 73.

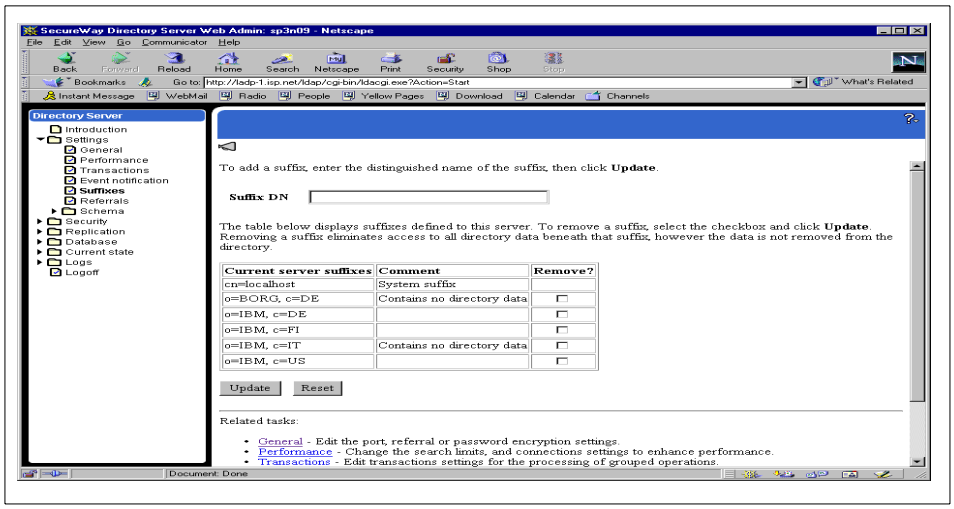


Figure 73. Checking the newly created suffixes

The global store for the LDAP Data Interchange Format (LDIF) data files was defined as /projects/LDAP. Thus, we created this directory and updated the automounter configuration, as described in Section 6.2.11, “Tailoring the NFS automounter” on page 235. The NFS exports file for node6 did not need to be updated, because the top level directory /export/projects was already exported

We changed the /cws/cust/cws01/auto.isp file and added the line:

`LDAP node6:/export/projects/LDAP`

To create the directory on node4 and update all available nodes, we ran the following command sequence on the CWS:

```
<cws01>#dsh -w node4 mkdir /export/projects/LDAP
<cws01>#dsh -w node4 chmod 755 /export/projects/LDAP
<cws01>#/cws/cust/common/AMD_entry.ksh
```

We created a sample LDIF file for testing purposes only called `cust1.ldif` within directory `/projects/LDAP` with the following contents:

```
dn: o=BORG, c=DE
objectclass: top
objectclass: organization
o: BORG

dn: ou=UNIMATRIX, o=BORG, c=DE
ou: UNIMATRIX
objectclass: organizationalUnit
seealso: cn=7of9, ou=UNIMATRIX, o=BORG, c=DE

dn: cn=7of9, ou=UNIMATRIX, o=BORG, c=DE
objectclass: person
objectclass: organizationalPerson
cn: 7of9
sn: Seven
telephonenumber: 49-812-855-5492
internationaliSDNNumber: 119-5492
title: Technical Team
```

Note

Do not use Common Names (CN) longer than eight characters. This is the maximum length of characters passed from a Web browser basic authentication dialog to the Web server.

To import the LDIF file into the DB2 database, we used the `ldifdb2` command delivered with SecureWay Directory Server:

```
<cws01>#dsh -w node6 /usr/ldap/sbin/ldif2db -i
1  /projects/LDAP/cust1.ldif
node6: Configuration read securePort 636.
node6: ldif2db: 3 entries have been successfully added out
1 of 3 attempted.
```

1 These lines have been split for redbook printing purposes. In the actual command and output sequence, they are on a single line.

To take all changes into effect, we needed to restart the LDAP server daemon slapd, either using the Web interface or using the `dsh` command. After that, we checked the new directory entries using the `ldapsearch` command line utility:

```
<cws01>#  dsh -w node6 ldapsearch -h
1 node6 -p 389 -b "o=BORG,c=DE" -s sub "objectclass=*"
node9: o=BORG, c=DE
node9: objectclass=top
node9: objectclass=organization
node9: o=BORG
node9:
node9: ou=UNIMATRIX, o=BORG, c=DE
node9: ou=UNIMATRIX
node9: objectclass=organizationalUnit
node9: objectclass=top
node9: seealso=cn=7of9, ou=UNIMATRIX, o=BORG, c=DE
node9:
node9: cn=7of9, ou=UNIMATRIX, o=BORG, c=DE
node9: objectclass=person
node9: objectclass=organizationalPerson
node9: objectclass=top
node9: cn=7of9
node9: sn=Seven
node9: telephonenumber=49-812-855-5492
node9: internationalisdnnumber=119-5492
node9: title=Technical Team
```

1 These lines have been split for redbook printing purposes. In the actual command and output sequence, they are on a single line.

6.4.5 Integrating LDAP as the basic authentication method for IHS

Next, we set up the sample Web server www.isp.net with LDAP authentication protected pages under URL `http://www.isp.net/protected/`. The IHS fileset http_server.modules.ldap has already been installed through our standard IHS installation, as described with Section 6.4.2, “Installing IBM HTTP Server” on page 262.

First, we changed the IHS basic configuration file http.conf that is located in the globalized directory /projects/www/IHSConf, and added the following lines:

```
LoadModule ibm_ldap_module     libexec/mod_ibm_ldap.so
AddModule mod_ibm_ldap.c
LdapConfigFile "/projects/www/IHSConf/wwwldap.prop"
<Directory /projects/www/html/protected>
```

```
order allow,deny
allow from all
AuthName       "LDAP Realm"
AuthType       Basic
Require                valid-user
LdapConfigFile "/projects/www/IHSconf/wwwldap.prop"
</Directory>
```

We then created the IHS LDAP properties file from the sample file /usr/HTTPServer/conf/ldap.prop.sample, which is located in the globalized directory /projects/www/IHSconf/ as wwwldap.prop.

In our example, the files read:

```
ldap.realm=LDAP Realm
ldap.URL=ldap://192.168.3.6:389/ou=UNIMATRIX,o=BORG,c=DE
ldap.application.authType=None
ldap.user.authType=BasicIfNoCert
```

We distributed and refreshed the Web server using the following script on the CWS, as described in Section 6.4.2, “Installing IBM HTTP Server” on page 262:

```
<cws01>#dsh -w node3 /cws/cust/common/IHS_www_update_server.ksh
<cws01>#dsh -w node4 /cws/cust/common/IHS_www_update_server.ksh
```

Finally, we tested the Web server behavior by accessing the URL `http://www-1.isp.net/protected`. The verification panel that appears is shown in Figure 74 on page 278.

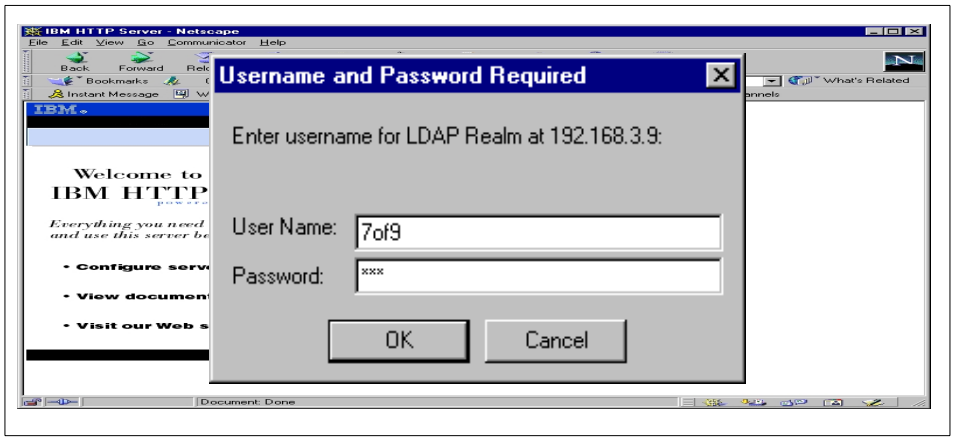


Figure 74. LDAP IHS verification login panel

6.5 Special topics

Here is another brief overview of the overall capabilities that the RS/6000 SP environment has in terms of daily business operations.

6.5.1 Working with the syslog log file

In most cases, debugging can be achieved using AIX syslog entries. A general way to select dedicated systems can be done using the general logging directory. In the example, we will show how to select dedicated systems for debugging:

1. We edit the nodes syslog.conf file located in the directory /cws/cust/node4/etc and add the line:

   ```
   *.debug   /cws/log/node4/syslog
   ```

2. Use the `dsh` command to update node4 environment:

```
<cws01>#dsh -w node4node4 cp /cws/cust/node4/etc/syslog.conf
1  /etc/syslog.conf
<cws01>#dsh -w node4 touch /cws/log/node4node4/syslog
<cws01>#dsh -w node4 refresh -s syslogd
node4: 0513-095 The request for subsystem refresh was
ompleted successfully.
<cws01>#tail -f /cws/log/node4/syslog
Oct 25 13:49:28 node4 automountd[12398]:
Attempt a TCP connection.
Oct 25 13:49:28 node4 automountd[12398]: TCP - Success!
Oct 25 13:50:57 node4 ftpd[3232]: <--- 220 node4
FTP server (Version wu-2.6.1(1) Thu Jul 20 19:10:14 DFT 2000)
ready.
Oct 25 13:50:59 node4 ftpd[3232]: command: USER root^M
Oct 25 13:50:59 node4 ftpd[3232]: <--- 331
Password required for root.
Oct 25 13:50:59 node4 ftpd[3232]: USER root
Oct 25 13:51:00 node4 ftpd[3232]: command: PASS password^M
Oct 25 13:51:00 node4 ftpd[3232]: PASS password
Oct 25 13:51:00 node4 ftpd[3232]: <--- 230 User
root logged in.
Oct 25 13:51:00 node4 ftpd[3232]: FTP LOGIN FROM www1.isp.net
[192.168.3.201], root
Oct 25 13:51:08 node4 ftpd[3232]: command: QUIT^M
Oct 25 13:51:08 node4 ftpd[3232]: QUIT
Oct 25 13:51:08 node4 ftpd[3232]:
<--- 221-You have transferred 0 bytes in 0 files.
Oct 25 13:51:08 node4 ftpd[3232]:
--- 221-Total traffic for this session was
206 bytes in 0 transfers.
Oct 25 13:51:08 node4 ftpd[3232]:
<--- 221-Thank you for using the FTP service on node4.
Oct 25 13:51:08 node4 ftpd[3232]: <--- 221 Goodbye.
Oct 25 13:51:08 node4 ftpd[3232]: FTP session closed
```

1 These lines have been split for redbook printing purposes. In the actual command and output sequence, they are on a single line.

3. To disable syslog logging, we comment the debug line:

   ```
   #*.debug  /cws/log/node4/syslog
   ```

   in the file /cws/cust/node4/etc/syslog.conf. We then copy the configuration file from the CWS to the /etc directory and refresh the daemon on node4:

```
<cws01>#dsh -w node4node4 cp /cws/cust/node4/etc/syslog.conf
1  /etc/syslog.conf
<cws01>#dsh -w node4 refresh -s syslogd
```

1 These lines have been split for redbook printing purposes. In the actual command and output sequence, they are on a single line.

6.5.2 Scheduled backup

Scheduled backups are essential for recovering from unexpected hardware failures. On the other hand, regular scheduled full system backups enable flexible deployment of available or new installed hardware.

The simple shell script called FSystemb.ksh generates a full system backup from the CWS of the node passed as an argument to the script:

```
#!/bin/ksh
#FSystemb.ksh created for redbook SG24-6025
TODATE=$(date +"%b_%d")
#
N=$1
CWS=sp3en0
DIR="/export/fsystemb/"
#set -x
#
if [ ! -d /${DIR}/${N} ]
then
 mkdir /${DIR}/${N}
 chmod 775 /${DIR}/${N}
fi
#
dsh -w ${N} umount /mnt
dsh -w ${N}  mount ${CWS}:/${DIR} /mnt
#
dsh -w ${N} mksysb -i /mnt/${N}/${N}.fsystmb${TODATE}
#
dsh -w ${N} umount /mnt
```

We added the following entries to CWS crontab file, which back ups node1 and node2 on an scheduled basis:

```
30 4 1,15 * * /home/root/cust/FSystemb.ksh node1
30 4 2,16 * * /home/root/cust/FSystemb.ksh node2
```

6.6 Closing statement

At this point, we have completed all the tasks for the sample ISP implementation based on the layout described in Section 5.2, “Market visibility model” on page 199 and Section 5.3, “Enhanced market visibility

model” on page 202. There is a lot of work left to transform this sample installation into a production environment.

We showed the reader how the RS/6000 SP Systems can manage the complete SP system as well as all the application specific tasks. All the management operations are centralized using the CWS as the focal point. The advantages of using the RS/6000 SP for an ISP implementation are described in Section 4.5, “Benefits of using the SP in the ISP arena” on page 193.

The RS/6000 SP functionality, combined with a secure distributed file system such as extended access control list, enables a wide range of opportunities in an ISP/ASP environment.

Appendix A. Network protocols

This appendix lists the definitions for the most common network protocols used.

HyperText Transport Protocol (HTTP)

HTTP is the TCP-based protocol used by World Wide Web applications, which enables the interaction between Web browsers and servers. HTTP is a stateless protocol designed to support hypermedia information transfer. It allows an application to take advantage of the CGI (Common Gateway Interface) and CCI (Common Client Interface).

The HTTP protocol is based on a request/response paradigm. The client establishes a connection with and sends a request to the server. The server sends the response and then closes the connection. A feature of this protocol is the typing and negotiation of data representation, allowing systems to be built independently of the data being transferred.

HTTPS

HTTPS is a secure version of HTTP using Secure Socket Layer (SSL) as a transport-layer security mechanism, which provides a secure pipe between the Web server and the Web client (browser).

WAP

The Wireless Application Protocol (WAP) is a set of protocols targeted at wireless data communications. The WAP protocols and architecture are designed to provide Web contents to mobile terminals, like mobile phones and communicators. WAP is designed to support many different terminal devices and different bearers. Wireless Markup Language (WML) a lightweight markup language, similar to HTML, but optimized for use in mobile terminals.

Internet Inter-ORB Protocol (IIOP)

If objects are to be oblivious to network connection, they interact merely via ORBs. CORBA specifications for object interoperability include two protocols:

- General Inter-ORB Protocol (GIOP)
- Environment Specific Inter-ORB Protocol (ESIOP)

The best known (and mandatory) implementation of GIOP is over TCP/IP transport, which is the Internet IOP (IIOP). The only ESIOP implementation is over DCE, the CIOP.

Java RMI

Java Remote Method Invocation (RMI) is a set of APIs designed to support remote method invocations on objects across Java virtual machines. RMI directly integrates a distributed object model into the Java language such that it allows distributed applications to be built in Java.

Java components that used RMI interface can make call on a remote object once it obtains a reference to the remote object, either by looking up the remote object in the bootstrap naming service provided by RMI or by receiving the reference as an argument or a return value. Java RMI uses a combination of Java Object Serialization and the Java Remote Method Protocol (JRMP), which convert normal-looking method calls into remote method calls.

File Transfer Protocol (FTP)

The FTP protocol provides a connection-oriented data transfer mechanism between client and server (the data transfer can be in either direction). The client must identify itself to the server, and the server is responsible for authenticating the client before it allows the file transfer.

FTP uses TCP as a transport protocol to provide reliable end-to-end connections. Two connections are used, one for control information and one for data.

Trivial File Transfer Protocol (TFTP)

The TFTP protocol provides a simple disk-to-disk data transfer mechanism. It is implemented using UDP and lacks most of the features of FTP (such as authentication).

Telnet

The TELNET protocol provides a standardized interface through which a program on one host (the TELNET client) may access the resources of another host (the TELNET server) as though the client were a local terminal connected to the server.

Chat

Internet Relay Chat (IRC) is a facility used by Internet users to chat with each other in real time. It is used mainly for recreational purposes, as it is very weak in terms of security. All IRC servers in the Internet are connected to each other. When an IRC client establishes a connection with a server, it then has access to all available channels for all servers. Users can use IRC client software, or telnet into the IRC server to use this facility.

News

UseNet is a discussion group or conferencing facility. Newsgroups are accessed through a client interface called a newsreader. While discussion groups implemented through mailing lists or mail posting to all subscribed users, UseNet keeps the news items at the servers, and clients retrieve them from there or read them directly from the server.

Network News Transfer Protocol (NNTP)

NNTP is the predominant protocol used by computers (servers and clients) for managing the notes posted on Usenet newsgroups. NNTP replaced the original Usenet protocol, UNIX-to-UNIX Copy Protocol (UUCP) some time ago. NNTP servers manage the network of collected Usenet newsgroups and include the server at your Internet access provider. The NNTP client may be included as part of your Netscape, Internet Explorer, Opera, or other Web browser or you may use a separate client program called a newsreader.

File Transfer Protocol (FTP)

The FTP protocol provides a connection-oriented data transfer mechanism between client and server. The data transfer can be in either direction. The client must identify itself to the server, and the server is responsible for authenticating the client before it allows the file transfer.

FTP uses TCP as a transport protocol to provide reliable end-to-end connections. Two connections are used, one for control information and one for data.

Appendix B. Sample configuration files

The sample configuration files used in Chapter 6, “Sample implementation” on page 219 are placed here. A file not shown in this section means that its contents has not been changed for the sample ISP integration.

B.1 Customized files for the basic operating system

The following files are the customized files for the basic OS.

B.1.1 rc.tcpip file

```
#! /bin/bsh#
# IBM_PROLOG_BEGIN_TAG
# This is an automatically generated prolog.
#
# tcpip42G src/tcpip/etc/rc.tcpip
#
# Licensed Materials - Property of IBM
#
# (C) COPYRIGHT International Business Machines Corp. 1985,1995
# All Rights Reserved
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# IBM_PROLOG_END_TAG
# @(#)95        1.62  src/tcpip/etc/rc.tcpip, tcpip, tcpip42G, g9701A
12/10/96 10:18:46
#
# COMPONENT_NAME: TCPIP rc.tcpip
#
# FUNCTIONS:
#
# ORIGINS: 26  27
#
# (C) COPYRIGHT International Business Machines Corp. 1985, 1996
# All Rights Reserved
# Licensed Materials - Property of IBM
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#

######################################################################
```

```
# rc.tcpip -
#       assumes interfaces are brought up by /etc/rc.net
#       starts TCP/IP daemons (sendmail, inetd, etc.)
######################################################################
# start -
#       starts daemons using either src or command-line method
# args:
#       $1: pathname of daemon
#       $2: non-null if we should use src to start the daemon
#       $3: any arguments to pass it
#
start()
{
        # just return if the daemon doesn't exist
        #
        [ -x $1 ] || return 0
# start the daemon using either src or command-line method
        #
        cmd=`basename $1`
        if [ -n "$2" ] ; then
                startsrc -s $cmd -a "$3"
        else
                if [ $cmd = "portmap" ] ; then
                        $1 $3 &  # portmap must start in background
                else
                        $1 $3
                fi
                echo "\t$cmd"
        fi
}
# check the bootup_option flag in the configuration database
option=`lsattr -E -l inet0 -a bootup_option -F value`
if [ "$option" = "no"  ]
then
######################################################################
#
# Check to see if srcmstr is running; if so, we try to use it;
# otherwise, we start the daemons without src
#
i=3  # make sure init has time to start it
while [ $i != 0 ] ; do
        if [ -n "`ps -e | awk '$NF == "srcmstr" { print $1; exit }'`" ] ;
then
                src_running=1  # set flag
                break
        fi
        i=`expr $i - 1`  # decrement count
```

```
done
# If srcmstr is running, ensure that it is active before issuing the
# startsrc commands
#
if [ -n "$src_running" ] ; then
        echo "Checking for srcmstr active...\c"
        i=10  # try ten times to contact it
        while [ $i != 0 ] ; do
                lssrc -s inetd >/dev/null 2>&1 && break  # break out on
success
                sleep 1  # otherwise wait a second and try again
                echo ".\c"
                i=`expr $i - 1`  # decrement count
        done
        if [ $i = 0 ] ; then
                echo "\n\nERROR: srcmstr is not accepting connections.\n"
                exit 1
        fi
        echo "complete"
fi
else
        src_running=""
fi
# Start up the daemons
#
echo "Starting tcpip daemons:"
trap "echo Finished starting tcpip daemons." 0
# Start up dhcpcd daemon
#start /usr/sbin/dhcpcd "$src_running"
# Start up syslog daemon (for error and event logging)
start /usr/sbin/syslogd "$src_running"
# Start up print daemon
#start /usr/sbin/lpd "$src_running"
# Start up routing daemon (only start ONE)
#start /usr/sbin/routed "$src_running" -q
#start /usr/sbin/gated "$src_running"
# Start up the sendmail daemon.
#
# Sendmail will automatically build the configuration and alias
# data bases the first time it is invoked.  You may wish to update
# Sendmail will automatically build the configuration and alias
# data bases the first time it is invoked.  You may wish to update
# the alias source file /usr/lib/aliases with local information,
# and then rebuild the alias data base by issuing the command
# "/usr/lib/sendmail -bi" or "/usr/ucb/newaliases".
#
# When the configuration or alias data bases are changed, the
```

```
# sendmail daemon can be made to rebuild and re-read them by
# issuing the command "kill -1 `cat /etc/sendmail.pid`" or, if
# SRC was used to start the daemon, "refresh -s sendmail".
#
# The "qpi", or queue processing interval, determines how
# frequently the daemon processes the message queue.
#
qpi=30m  # 30 minute interval
#
#start /usr/lib/sendmail "$src_running" "-bd -q${qpi}"
# Start up Portmapper
#start /usr/sbin/portmap "$src_running"
# Start up socket-based daemons
start /usr/sbin/inetd "$src_running"
# Start up Domain Name daemon
#start /usr/sbin/named "$src_running"
# Start up time daemon
#start /usr/sbin/timed "$src_running"
# Start up Network Time Protocol (NTP) daemon
#start /usr/sbin/xntpd "$src_running"
# Start up rwhod daemon (a time waster)
#start /usr/sbin/rwhod "$src_running"
# Start up the Simple Network Management Protocol (SNMP) daemon
#start /usr/sbin/snmpd "$src_running"
# Start up the DHCP Server
#start /usr/sbin/dhcpsd "$src_running"
# Start up the DHCP Relay Agent
#start /usr/sbin/dhcprd "$src_running"
# Start up the DPID2 daemon
#start /usr/sbin/dpid2 "$src_running"
# Start up the mrouted daemon
#start /usr/sbin/mrouted "$src_running"
```

B.1.2 ined.conf file

```
## @(#)62       1.17.1.8  src/tcpip/etc/inetd.conf, tcpip, tcpip420, 9613T
12/15/95 11:16:05
## IBM_PROLOG_BEGIN_TAG
## This is an automatically generated prolog.
##
## tcpip420 src/tcpip/etc/inetd.conf
##
## Licensed Materials - Property of IBM
##
## (C) COPYRIGHT International Business Machines Corp. 1993,1996
## All Rights Reserved
##
```

```
## US Government Users Restricted Rights - Use, duplication or
## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
##
## IBM_PROLOG_END_TAG
##
## COMPONENT_NAME: TCPIP inetd.conf
##
## FUNCTIONS:
##
## ORIGINS: 26  27
##
## (C) COPYRIGHT International Business Machines Corp. 1993
## All Rights Reserved
## Licensed Materials - Property of IBM
##
## US Government Users Restricted Rights - Use, duplication or
## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
##
## service  socket  protocol  wait/  user    server    server program
##  name     type             nowait         program     arguments
##
bootps  dgram   udp     wait    root    /usr/sbin/bootpd       bootpd
/etc/bootptab
tftp     dgram  udp6    SRC     nobody  /usr/sbin/tftpd         tftpd -n
instsrv stream  tcp     nowait  netinst /home/netinst/bin/instsrv instsrv
-r /tmp/netinstalllog /home/netinst/scripts
kshell  stream  tcp     nowait  root    /usr/sbin/krshd krshd
klogin  stream  tcp     nowait  root    /usr/sbin/krlogind krlogind
xmquery dgram   udp     wait    root    /usr/bin/xmservd xmservd -p3
spseccfg stream  tcp     nowait  root     /usr/lpp/ssp/bin/spseccfg spseccfg
switchtbld    stream  tcp     nowait  root    /usr/bin/switchtbld switchtbld
kfcli stream tcp nowait root /usr/lpp/ssp/install/bin/kfserver kfserver
```

B.1.3 Services file

```
DB2_db2inst1 60000/tcp   # Connection port for DB2 instance
DB2_db2inst1a   60001/tcp   # Interrupt  port for DB2 instance
db2cdb2inst1    50000/tcp   # Connection port for DB2 instance db2inst1
db2idb2inst1    50001/tcp   # Interrupt  port for DB2 instance
db2inst1DB2_ldapdb2     60002/tcp   # Connection port for DB2 LDAP instance
```

B.1.4 Aliases file

```
# @(#)87        1.3  src/bos/usr/lib/sendmail/aliases, cmdsend, bos420,
9613T 6/15/90 23:21:43
# IBM_PROLOG_BEGIN_TAG
# This is an automatically generated prolog.
```

```
#
# bos420 src/bos/usr/lib/sendmail/aliases
#
# Licensed Materials - Property of IBM
#
# (C) COPYRIGHT International Business Machines Corp. 1985,1989
# All Rights Reserved
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# IBM_PROLOG_END_TAG
#
# COMPONENT_NAME: CMDSEND aliases
#
# FUNCTIONS:
#
# ORIGINS: 10  26  27
#
# (C) COPYRIGHT International Business Machines Corp. 1985, 1989
# All Rights Reserved
# Licensed Materials - Property of IBM
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
##
#  Aliases in this file will NOT be expanded in the header from
#  Mail, but WILL be visible over networks or from /bin/bellmail.
#
#       >>>>>>>>>>      The command "sendmail -bi" must be run after
#       >> NOTE >>      this file is updated for any changes to
#       >>>>>>>>>>      affect sendmail operation.
##

# Alias for mailer daemon
MAILER-DAEMON:root

# Following alias is required by the new mail protocol, RFC 822
postmaster:root

# Aliases to handle mail to msgs and news
nobody: /dev/null
#added for Redbook SG24-6025
ssa_adm: root
root: root@node4.sp2.int
```

B.1.5 resolv.conf file

```
nameserver      10.30.0.1
nameserver      10.30.0.2
domain  isp.net
```

B.1.6 netsvc.conf file

```
hosts=local,bind4
```

B.1.7 Hosts file

```
# @(#)47        1.1  src/bos/usr/sbin/netstart/hosts, cmdnet, bos430,
9737A_430 7/24/91 10:00:46
# IBM_PROLOG_BEGIN_TAG
# This is an automatically generated prolog.
#
# bos430 src/bos/usr/sbin/netstart/hosts 1.1
#
# Licensed Materials - Property of IBM
#
# (C) COPYRIGHT International Business Machines Corp. 1985,1989
# All Rights Reserved
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# IBM_PROLOG_END_TAG
#
# COMPONENT_NAME: TCPIP hosts
#
# FUNCTIONS: loopback
#
# ORIGINS: 26  27
#
# (C) COPYRIGHT International Business Machines Corp. 1985, 1989
# All Rights Reserved
# Licensed Materials - Property of IBM
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
#  /etc/hosts
#
# This file contains the hostnames and their address for hosts in the
# network.  This file is used to resolve a hostname into an Internet
# address.
```

```
#
# At minimum, this file must contain the name and address for each
# device defined for TCP in your /etc/net file.  It may also contain
# entries for well-known (reserved) names such as timeserver
# and printserver as well as any other host name and address.
#
# The format of this file is:
# Internet Address      Hostname        # Comments
# Items are separated by any number of blanks and/or tabs.  A '#'
# indicates the beginning of a comment; characters up to the end of the
# line are not interpreted by routines which search this file.  Blank
# lines are allowed.
# Internet Address      Hostname        # Comments
# 192.9.200.1           net0sample      # ethernet name/address
# 128.100.0.1           token0sample    # token ring name/address
# 10.2.0.2              x25sample       # x.25 name/address
127.0.0.1             loopback localhost     # loopback (lo0) name/address
#written for Redbook SG24-6025
# SP Ethernet
10.30.0.1       node1.sp2.int  node1
10.30.0.2       node2.sp2.int  node2
10.30.0.3       node3.sp2.int  node3
10.30.0.4       node4.sp2.int  node4
10.30.0.5       node5.sp2.int  node5
10.30.0.6       node6.sp2.int  node6
#
10.30.0.254     cws01.sp2.int  cws01
#
```

B.2 BOS and user environment files

The following files are sample user environment files.

B.2.1 Profile file

```
#!/bin/ksh
#-------------------------------------------------------------------#
#  File   : /tftpboot/.profile.node                                 #
#                                                                   #
#  Author : L. Denefleh                                             #
#  Date   : 10/06/2000  (dd/mm/yyyy)                                #
#  Update :             (dd/mm/yyyy)                                #
#  What   : global profile                                          #
#                                                                   #
#  Abstract : this is the user profile executed after login         #
```

```
#-------------------------------------------------------------------#
# START The KORN-Shell                                              #
#-------------------------------------------------------------------#
START=$HOME/.kshrc ; export START
ENV='${START[(_$-=1)+(_=0)-(_$-!=_${-%%*i*})]}'; export ENV
PATH=$PATH:/usr/lpp/ssp/bin:/usr/lpp/ssp/kerberos/bin/
set -o vi
VISUAL=/usr/bin/vi
if [ -s "$MAIL" ]
    then echo "$MAILMSG"
fi
#-------------------------------------------------------------------#
# EOF                                                               #
#-------------------------------------------------------------------#
```

B.2.2 kshrc file

```
#!/bin/ksh
#set -x
#===================================================================#
#  File   : /tftpboot/.kshrc.node                                   #
#                                                                   #
#  subject: global .kshrc for root user                             #
#  usgae  : ln -s                                                   #
#  Date   : 10/06/2000  (dd/mm/yyyy)                                #
#  Update :             (dd/mm/yyyy)                                #
#                                                                   #
#===================================================================#
#     Use the extended commands of the k-shell                      #
#-------------------------------------------------------------------#

set -o ignoreeof

logname=`id | cut -d'(' -f2 | cut -d')' -f1`

if [ "$logname" = "root" ]
then
   prompt='#'
else
   prompt='$'
fi
export prompt

chan=`tty` ; export chan
TTY=`basename $chan` ; export TTY
path=`pwd`; export path
NODE=`hostname | cut -d"." -f1`; NODE=$logname"@"$NODE; export NODE
```

```
if [ "$TTY" = "console" ]
    then PS1='[$NODE:]$PWD $prompt '
    else PS1='[$TTY:$NODE:]$PWD $prompt '
fi
EDITOR=/usr/bin/vi
FCEDIT=/usr/bin/e
CDPATH=:
VISUAL=vi ; export VISUAL
export PS1 EDITOR FCEDIT CDPATH
HISTFILE=$HOME/.sh_history
HISTSIZE=1024
set -o monitor
#------------------------------------------------------------------#
# EOF                                                              #
#------------------------------------------------------------------#
```

B.2.3 motd file

```
*************************************************************************
*                                                                       *
*                                                                       *
*  Welcome to the Internet AIX System used developing examples for *
*  Redbook SG24-6025 Integrating an ISP into an RS/6K SP2 System *
*                                                                       *
*                                                                       *
*  Please see the README file in /usr/lpp/bos for information pertinent to*
*  this release of the AIX Operating System. *
*                                                                       *
*                                                                       *
* Last Update 10/06/20000                        need help: root@isp.net *
*************************************************************************
```

B.2.4 Environment file

```
PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:/usr/bin/X11:/sbin
TZ=EST5EDT
LANG=en_US
LOCPATH=/usr/lib/nls/loc
NLSPATH=/usr/lib/nls/msg/%L/%N:/usr/lib/nls/msg/%L/%N.cat
LC__FASTMSG=true
#
JAVA_HOME=/usr/jdk_base
NSORDER=local,bind4
```

B.3 Network Dispatcher configuration files and sample listing

The following files are sample Network Dispatcher configuration files.

B.3.1 start.cfg file for backup eND server

```
#!/bin/ksh
#set -x
#Backup nd startup script written for  Redbook SG24-6025
ndserver start
ndcontrol executor start
NFA=192.168.3.2
CLUSTER=192.168.3.200
ndcontrol executor set nfa $NFA
ndcontrol cluster add $CLUSTER
ndcontrol port add $CLUSTER:20+21+80
SERVER1=192.168.3.3
SERVER2=192.168.3.4
ndcontrol server add $CLUSTER:20+21+80:$SERVER1+$SERVER2
ndcontrol manager start
ndcontrol advisor start ftp 21
ndcontrol advisor start http 80
ndcontrol manager proportions 58 40 2 0
#
ndcontrol set loglevel 5
ndcontrol set logsize unlimited
ndcontrol manager loglevel 5
#
ndcontrol highavailability heartbeat add 192.168.3.2 192.168.3.1
ndcontrol highavailability reach add 192.168.3.130
ndcontrol highavailability backup add backup manual 11001
```

B.3.2 goActive file

```
#!/bin/ksh
#
# goActive script
#
# Configure this script when using the high availability feature of
# Network Dispatcher.
#
# This script is executed when Network Dispatcher goes into the
# 'Active' state and begins routing packets.
#
# This script must be placed in Network Dispatcher's bin directory (by default
# this is /usr/lpp/nd/dispatcher/bin) and it needs to have root execute permission.
#
# Modify NETWORK, CLUSTER, INTERFACE and NETMASK to match your environment.
#
# en0=first Ethernet adapter, tr0=first Token ring adapter, fi0=first FDDI adapter.
#
# NETMASK must be the netmask of your LAN.  It may be hexadecimal or dotted-decimal
notation.
#
HOST=$(hostname)
TODAY="$(date +"%d %m %y %T")"
#
#ND_LOGDIR=/usr/lpp/nd/dispatcher/logs
NETWORK=192.168.3
INTERFACE=en0
```

```
#NETMASK=255.255.255.0
NETMASK=0xffffff00
#
#
#
#   date >> $ND_LOGDIR/ha.log
#   print "This machine is Active.  Aliasing cluster address(es) to NIC \n" >>
$ND_LOGDIR/ha.log
  for CLUSTER in 11 12; do
     ifconfig lo0 delete $NETWORK.$CLUSTER
     ifconfig $INTERFACE alias $NETWORK.$CLUSTER netmask $NETMASK
  done
#
/usr/bin/mail -s "${HOST} NDisptacher gone active ${TODAY}" root@isp.net.com < /dev/null
```

B.3.3 goInOp file

```
#!/bin/ksh
#
# goInOp script
#
# Configure this script when using the high availability feature of
# Network Dispatcher and optionally when using Network Dispatcher in a
# standalone environment.
#
# This script is executed when the Network Dispatcher executor is stopped
# (and before the executor is initially started).
#
# This script must be placed in Network Dispatcher's bin directory
# (nd/dispatcher/bin/) and it needs to have root execute permission.
#
# Modify NETWORK, CLUSTER and INTERFACE to match your environment.
#
# en0=first Ethernet adapter, tr0=first Token ring adapter, fi0=first FDDI adapter
#
#ND_LOGDIR=/usr/lpp/nd/dispatcher/logs
NETWORK=192.168.3
INTERFACE=en0
#
#   date >> $ND_LOGDIR/ha.log
#   print "Executor has stopped.  Removing loopback and device aliases. \n" >>
$ND_LOGDIR/ha.log
 for CLUSTER in 11 12; do
   ifconfig lo0 delete $NETWORK.$CLUSTER
   ifconfig $INTERFACE delete $NETWORK.$CLUSTER
 done
#
```

B.3.4 goStandby file

```
#!/bin/ksh
#
# goStandby script
#
# Configure this script when using the high availability feature of
# Network Dispatcher.
#
# This script is executed when Network Dispatcher goes into the
# 'Standby' state.  Monitoring the health of the 'Active' machine
# but not routing packets.
#
```

```
# This script must be placed in Network Dispatcher's bin directory
# (nd/dispatcher/bin/) and it needs to have root execute permission.
#
# Modify NETWORK, CLUSTER, INTERFACE and NETMASK to match your environment.
#
# en0=first Ethernet adapter, tr0=first Token ring adapter, fi0=first FDDI adapter
#
# NETMASK must be the netmask of your LAN.  It may be hexadecimal or octal notation.
#
HOST=$(hostname)
TODAY="$(date +"%d %m %y %T")"
#
#ND_LOGDIR=/usr/lpp/nd/dispatcher/logs
NETWORK=192.168.3
INTERFACE=en0
#NETMASK=255.255.255.0
NETMASK=0xffffff00
#
#
#  date >> $ND_LOGDIR/ha.log
#  print "Going into Standby mode.\n" >> $ND_LOGDIR/ha.log
#  print "Deleting the device aliases and adding the loopback aliases" >>
$ND_LOGDIR/ha.log
 for CLUSTER in  11 12; do
    ifconfig $INTERFACE delete $NETWORK.$CLUSTER
    ifconfig lo0 alias $NETWORK.$CLUSTER netmask $NETMASK
 done
#
/usr/bin/mail -s "${HOST} NDisptacher gone StandBy ${TODAY}" root@isp.net <  /dev/null
#
```

B.3.5 Sample output from eND command ndcontrol manager report

```
---------------------------------
|  HOST TABLE LIST  |   STATUS   |
---------------------------------
|       192.168.3.3|      ACTIVE|
|       192.168.3.4|      ACTIVE|
---------------------------------
------------------------------------------------------------------------------------------
------------
|   192.168.3.200|   WEIGHT  |   ACTIVE %  58   |    NEW %  40    |   PORT %   2  | SYSTEM
%   0 |
------------------------------------------------------------------------------------------
------------
| PORT:    80    | NOW | NEW |  WT  |  CONNECT  |  WT  |  CONNECT  |  WT  |  LOAD  |  WT
|  LOAD  |
------------------------------------------------------------------------------------------
------------
|     192.168.3.3|  10 |  10 |   10 |         0 |   10 |         1 |   10 |    574 |    0
|      0 |
|     192.168.3.4|  10 |  10 |   10 |         0 |   10 |         0 |-9999 |     -1 |    0
|      0 |
------------------------------------------------------------------------------------------
------------
| PORT TOTALS:   |  20 |  20 |      |        0 |      |        1 |      |   573 |      |
0 |
------------------------------------------------------------------------------------------
------------
------------------------------------------------------------------------------------------
------------
```

```
|   192.168.3.200|   WEIGHT  |   ACTIVE %  58   |     NEW %  40    |   PORT %   2  | SYSTEM
%   0 |
-------------------------------------------------------------------------------------------
------------
| PORT:    21    | NOW | NEW |  WT  |  CONNECT  |  WT  |  CONNECT  |  WT  |  LOAD  |  WT
|  LOAD  |
-------------------------------------------------------------------------------------------
------------
|     192.168.3.3|  10 |  10 |   10 |         0 |   10 |         0 |-9999 |     -1 |    0
|      0 |
|     192.168.3.4|  10 |  10 |   10 |         0 |   10 |         0 |-9999 |     -1 |    0
|      0 |
-------------------------------------------------------------------------------------------
------------
| PORT TOTALS:   | 20 |  20 |      |        0 |      |        0 |      |    -2 |      |
0 |
-------------------------------------------------------------------------------------------
------------
-------------------------------------------------------------------------------------------
------------
|   192.168.3.200|   WEIGHT  |   ACTIVE %  58   |     NEW %  40    |   PORT %   2  | SYSTEM
%   0 |
-------------------------------------------------------------------------------------------
------------
| PORT:    20    | NOW | NEW |  WT  |  CONNECT  |  WT  |  CONNECT  |  WT  |  LOAD  |  WT
|  LOAD  |
-------------------------------------------------------------------------------------------
------------
|     192.168.3.3|   9 |   9 |   10 |         0 |   10 |         0 |    0 |      0 |    0
|      0 |
|     192.168.3.4|   9 |   9 |   10 |         0 |   10 |         0 |    0 |      0 |    0
|      0 |
-------------------------------------------------------------------------------------------
------------
| PORT TOTALS:   | 18 |  18 |      |        0 |      |        0 |      |     0 |      |
0 |
-------------------------------------------------------------------------------------------
------------
---------------------------------
| ADVISOR |  PORT  |  TIMEOUT  |
---------------------------------
|   http  |    80  | unlimited |
|    ftp  |    21  | unlimited |
---------------------------------
```

B.4 Node dependant services startup configuration files

The following files are sample node dependant services startup configuration files.

B.4.1 node1 rc.include file

```
#!/bin/ksh
#written for Redbook SG24-6025
#
/usr/bin/mknotify -n sendmail -m /cws/cust/common/restart_sendmail.ksh
/cws/cust/common/restart_sendmail.ksh
```

```
#
/usr/bin/mknotify -n named -m /cws/cust/common/restart_named.ksh
/cws/cust/common/restart_named.ksh
#
/usr/lpp/nd/dispatcher/bin/start.cfg
```

B.4.2 node2 rc.include file

```
#!/bin/ksh
#written for Redbook SG24-6025
#
/usr/bin/mknotify -n sendmail -m /cws/cust/common/restart_sendmail.ksh
/cws/cust/common/restart_sendmail.ksh
#
/usr/lpp/nd/dispatcher/bin/start.cfg
```

B.4.3 node3 rc.include file

```
#!/bin/ksh
#written for Redbook SG24-6025
#
/etc/ifconfig lo0 alias 192.168.3.200 netmask 255.255.255.0
#
/usr/HTTPServer/bin/httpd
#
```

B.4.4 node4 rc.include file

```
#!/bin/ksh
#written for Redbook SG24-6025
#
/etc/ifconfig lo0 alias 192.168.3.200 netmask 255.255.255.0
#
/usr/HTTPServer/bin/httpd
#
```

B.4.5 node6 rc.include file

```
#!/bin/ksh
#written for Redbook SG24-6025
#
/bin/slapd > /cws/log/${HOSTNAME}/slapd.out
#
/usr/HTTPServer/bin/httpd
#
```

Appendix C. Special notices

This publication is intended to help IBM customers, IBM business partners, IBM sales professionals, IBM I/T specialists, and IBM technical support team when proposing an RS/6000 SP based solution for an Internet Service Provider. The information in this publication is not intended as the specification of any programming interfaces that are provided by RS/6000 hardware, AIX software, or PSSP software. See the PUBLICATIONS section of the IBM Programming Announcement for RS/6000 for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee

that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

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

AS/400	Domino
IBM ®	Lotus
Lotus Notes	Notes
Redbooks	Redbooks Logo 
SecureWay	SP
SP2	System/390
VideoCharger	WebSphere
Wizard	XT

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere.,The Power To Manage., Anything. Anywhere.,TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

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

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

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

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

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

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

Appendix D. Related publications

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

D.1 IBM Redbooks

For information on ordering these publications see “How to get IBM Redbooks” on page 311.

- *AIX 4.3 Elements of Security: Effective and Efficient Implementation*, SG24-5962
- *An Introduction to IBM WebSphere Everyplace Suite Version 1.1 Accessing Web*, SG24-5995
- *DB2 UDB Installation and Configuration Supplement V6*, GC09-2857
- *Exploiting RS/6000 SP Security: Keeping it Safe*, SG24-5521
- *IBM WebSphere Trascoding Publisher V1.1: Extending Web Applications to the Pervasive World*, SG24-5965
- *Inside the RS/6000 SP*, SG24-5145
- *PSSP Version 3 Survival Guide*, SG24-5344
- *The RS/6000 SP Inside Out*, SG24-5374
- *RS/6000 SP Monitoring: Keeping it Alive*, SG24-4873
- *RS/6000 SP System Performance Tuning Update*, SG24-5340

D.2 IBM Redbooks collections

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at ibm.com/redbooks for information about all the CD-ROMs offered, updates and formats.

CD-ROM Title	Collection Kit Number
IBM System/390 Redbooks Collection	SK2T-2177
IBM Networking Redbooks Collection	SK2T-6022
IBM Transaction Processing and Data Management Redbooks Collection	SK2T-8038
IBM Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
IBM AS/400 Redbooks Collection	SK2T-2849
IBM Netfinity Hardware and Software Redbooks Collection	SK2T-8046
IBM RS/6000 Redbooks Collection	SK2T-8043
IBM Application Development Redbooks Collection	SK2T-8037

CD-ROM Title	Collection Kit Number
IBM Enterprise Storage and Systems Management Solutions	SK3T-3694

D.3 Other resources

- *AIX Version 4.3 System Management Guide: Communications and Networks*, SC23-4127
- *AIX Version 4.3 System Management Guide: Operating Systems and Devices*, SC23-4126
- *PSSP: Diagnosis Guide*, GA22-7350
- *PSSP: Event Management Programming Guide and Reference*, SC23-3996
- *PSSP: Group Services Programming Guide and Reference*, SC28-1675
- *PSSP: Installation and Migration Guide*, GA22-7347

To contact the ISV Center for NetGen for more information on non IBM products, use the following contact information:
Notes ID (IBM intranet only): EMEA ISVCenterForNetGen/France/IBM
Internet: ISVCTRNG@fr.ibm.com

D.4 Referenced Web sites

These Web sites are also relevant as further information sources:

- `http://www.ibm.com/servers/eserver/pseries/hardware/factsfeatures.html` - For further information on available type of nodes and attached servers
- `http:// www.ibm.com/e-business/infrastructure/uk/zones` - For further information on the description of the seven zones of an e-infrastructure
- `http://www-3.ibm.com/pvc/tech/nokiawap.shtml` - For further information on Nokia WAP Gateway
- `http://www-4.ibm.com/software/security/firewall/` - For further information on SecureWay Firewall software product
- `http://www-4.ibm.com/software/data/db2/` - For further information on DB2 software products
- `http://www-4.ibm.com/software/webservers/httpservers` - For further information on IBM HTTP Server software product

- http://www-4.ibm.com/software/webservers/appserv - For further information on Websphere Administrative Console
- http://www-4.ibm.com/software/webservers/edgeserver - For further information on the Edge Server Caching Proxy features
- http://www-4.ibm.com/software/ts/mqseries/everyplace - For further information on MQSeries Everyplace
- http://www-4.ibm.com/software/data/cm - For further information on content management workflow with MQSeries Workflow
- http://maps.vix.com/rbl - For further information on IMS email server solution features
- http://www.bull.us.com - For details on freeware software products offered by Bull
- http://www.checkpoint.com - For further information on Firewall-1 software product
- http://www.citrix.com - For further information on Metaframe software products
- http://www.dns.net/dnsrd/tools.html - For further information on tools to implement Domain Name Systems (DNS)
- http://www.eecis.udel.edu/~ntp - For further information on Network Time Protocol (NTP)
- http://www.gtl.com - For further information on Geneva software product
- http://www.portal.com - For further information on Infranet software product
- http://www.tivoli.com/products - For further information on Tivoli products
- http://www.webopedia.com - For details on terminology definitions
- http://www.wu-ftpd.org - For further details on WU-FTP software

How to get IBM Redbooks

This section explains how both customers and IBM employees can find out about IBM Redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** ibm.com/redbooks

  Search for, view, download, or order hardcopy/CD-ROM Redbooks from the Redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

  Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

  Send orders by e-mail including information from the IBM Redbooks fax order form to:

	e-mail address
In United States or Canada	pubscan@us.ibm.com
Outside North America	Contact information is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	Country coordinator phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Fax Orders**

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	Fax phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the Redbooks Web site.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and Redbooks by accessing the IBM Intranet Web site at http://w3.itso.ibm.com/ and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access MyNews at http://w3.ibm.com/ for redbook, residency, and workshop announcements.

IBM Redbooks fax order form

Please send me the following:

Title	Order Number	Quantity

First name	Last name	
Company		
Address		
City	Postal code	Country
Telephone number	Telefax number	VAT number

☐ Invoice to customer number _____

☐ Credit card number _____

Credit card expiration date	Card issued to	Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

Abbreviations and acronyms

AIX	Advanced Interactive Executive
ASP	Application Service Provider
BOS	Base Operating System
BSS	Business Support System
CDR	Call Details Record
CHAP	Challenge-Handshake Authentication Protocol
CORBA	Common Object Request Broker Architecture
CRM	Customer Relationship Management
CSR	Customer Service Representative
CWS	RS/6000 SP Control Workstation
DCE	Distributed Computing Environment
DFS	Distributed File System
DMZ	De-Militarized Zone
DNS	Domain Name System
EAP	Extensible Authentication Protocol
FTP	File Transfer Protocol
HACMP	High Availability Cluster Multi-Processing
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
IDL	Interface Definition Language
IHS	IBM HTTP Server
IXC	Internet eXChange
IIOP	Internet Inter-ORB Protocol
IP	Internet Protocol
IPDR	Internet Protocol Details Record
IPSec	IP Security
IPX	Internetwork Packet Exchange
ISP	Internet Service Provider
iTk	Integration Tool Kit
JDBC	Java Data Base Connectivity
JFS	Journaled File System
JSP	Java Server Page
LDAP	Lightweight Directory Access Protocol
LDIF	LDAP Data Interchange Format
LVM	Logical Volume Manager
LPP	Licensed Software Product
NAS	Network Access Server
NFS	Network File System
ND	Network Dispatcher
NIM	Network Install Manager
NTP	Network Time Protocol
ORB	Object Request Broker
OSS	Operation Support System
PAP	Password Authentication Protocol
PDK	Portal Development Kit

PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PSSP	Parallel System Support Programs
PRE	Portal Runtime Environment
PTT	Post Telegraphy Telecom
QoS	Quality of Service
RADIUS	Remote Authentication Dial In User Service
RAID	Redundant Array of Independent Disks
RAS	Remote Access Server
RBOC	Regional Bell Operating Companies
RDBMS	Relational Data Base Management System
RMI	Remote Method Invocation
SMS	Short Message Service
SMSC	SMS Centre
SPX	Sequenced Packet Exchange
SSA	Serial Storage Architecture
SSL	Secure Socket Layer
TCB	Trusted Computing Base
TCP	Transmission Control Protocol
TiSM	Tivoli Subscription Manager
TLS	Transport Layer Security
UM	Unified Messaging
URL	Uniform Resource Locator
VG	Volume Group
VPN	Virtual Private Network
WAP	Wireless Application Protocol
WML	Wireless Markup Language
WSP	Wireless Session Protocol
WTLS	Wireless Transport Layer Security

Index

Symbols

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

IBM Redbooks review

Your feedback is valued by the Redbook authors. In particular we are interested in situations where a Redbook "made the difference" in a task or problem you encountered. Using one of the following methods, **please review the Redbook, addressing value, subject matter, structure, depth and quality as appropriate.**

- Use the online **Contact us** review redbook form found at **ibm.com**/redbooks
- Fax this form to: USA International Access Code + 1 845 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Document Number **Redbook Title**	SG24-6025-00 Integrating an ISP into a RS/6000 SP Environment
Review	
What other subjects would you like to see IBM Redbooks address?	
Please rate your overall satisfaction:	O Very Good O Good O Average O Poor
Please identify yourself as belonging to one of the following groups:	O Customer O Business Partner O Solution Developer O IBM, Lotus or Tivoli Employee O None of the above
Your email address: The data you provide here may be used to provide you with information from IBM or our business partners about our products, services or activities.	O Please do not use the information collected here for future marketing or promotional contacts or other communications beyond the scope of this transaction.
Questions about IBM's privacy policy?	The following link explains how we protect your personal information. **ibm.com**/privacy/yourprivacy/

Integrating an ISP into a RS/6000 SP Environment

Integrating an ISP into a RS/6000 SP Environment

Overview of what an Internet Service Provider is

Grow your ISP with a RS/6000 SP

Sample implementation, tips and techniques

This redbook takes you partly inside the Internet Service Provider (ISP) business and environment. It delivers an overview on what is an ISP, what are the technologies in place and how they work, what is coming in the near future, and how it is affecting the ISP business.

The redbook focuses on integrating an ISP environment into an RS/6000 SP environment. It identifies and implements the necessary hardware and software that are required for setting up an ISP configuration. This redbook describes the initial sizing, configuration guidelines, and step-by-step procedures for installation, mail configuration, and news, web and directory services that are required for an ISP solution.

The introduction defines the basic rules of the ISP market. It then defines the architecture and components for an ISP platform. Three solutions sets are proposed, and possible growing paths are described.

This redbook does not replace the latest RS/6000 marketing materials and tools. It is intended as an additional source of information that, together with existing resources, may be used to enhance your knowledge of IBM solutions for the UNIX marketplace.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

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

For more information:
ibm.com/redbooks

SG24-6025-00 ISBN 0738421022