Sun™ Internet Mail Server™ 4.0 Concepts Guide

THE NETWORK IS THE COMPUTER™

A Sun Microsystems, Inc. Business
901 San Antonio Road
Palo Alto, CA 94303 USA
650-960-1300 fax 650-969-9131

Part No.:805-7675-10
Revision A, July 1999

Contents

Figures

Tables

Preface

The *Sun Internet Mail Server 4.0 Concepts Guide* provides SIMS system administrators with conceptual understanding of the SIMS product. By understanding how SIMS works on a conceptual level, the administrator will be able to install the system using the *Sun Internet Mail Server 4.0 Installation Guide.* The administrator will also understand the administrative tasks that are described in the *Sun Internet Mail Server 4.0 Administrator's Guide*.

Topics in this chapter include:

- Who should use this book
- Conventions
- Related documentation
- Ordering Sun documents
- Shell prompts in command examples
- Graphical user interface conventions
- Notice

Who Should Use This Book

The intended audience for this installation guide is system administrators who are moderately experienced with managing a network of Sun Workstations™, PCs, Apple computers, or IBM mainframes. Previous experience in planning, installing, configuring, maintaining, and troubleshooting an enterprise email system also helps to efficiently use this guide.

Conventions

The following table describes the typographic conventions used in this book.

TABLE P-1 Typographic Conventions

Convention	Meaning	Example
`courier font`	Names of commands, files, on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% test.doc.`
italics	Book titles, new terms, words to be emphasized, variables that you replace with a real value	Read Chapter 6 in *User's Guide*. These are called *class* options. You *must* be root to do this. Type `rm` *filename* to delete a file.

`boldface courier font`	What you type	`machine_name%` **`su`**
		`Password:`

Related Documentation

SIMS 4.0 Documentation Set

The *Sun Internet Mail Server 4.0 Concepts Guide* manual is a prerequisite document for the following manuals in the SIMS documentation set:

Sun Internet Mail Server 4.0 Installation Guide – Describes the planning and installation procedures for the Sun Internet Mail Server (SIMS) 3.5 software on Solaris SPARC and Intel-based x86 systems. In particular, it describes the installation of the software using the Graphical User Interface (GUI).

Sun Internet Mail Server 4.0 Provisioning Guides—describes how to provision the SIMS LDAP directory with users, distribution lists, administrators, and domains by creating and importing LDIF records.

Sun Internet Mail Server 4.0 Administrator's Guide—describes how to fine-tune the default configuration, and maintain, monitor, and troubleshoot your mail server by using the SIMS Administration Console, a graphical user interface tool (GUI).

Sun Internet Mail Server 4.0 Reference Manual—Provides detailed information on command-line options, configuration files that can be edited by the administrator, system architecture, supported standards, and location of software files.

Sun Internet Mail Server Delegated Management Guide—describes the SIMS Delegated Management console and the tasks associated with the console. In particular, it describes how a delegated administrator for a hosted domain performs tasks on users and distribution lists.

Sun Internet Mail Server 4.0 Web Access Administrator's Guide—Describes the core system administration tasks for Sun Web Access software.

Reference Manual Pages (man pages)—Describe command-line utilities and detailed information about the arguments and attributes relevant to each command.

Sun Internet Mail Server 4.0 Release Notes—Covers open issues and late-breaking installation, administration, and reference information that is not published in the product books.

Other Related Sun Documentation

*Sun Internet Mail Server 4.0 Web site (*located at http://www.sun.com/sims*)*—Offers up-to-date information on a variety of topics, including: online product documentation and late-breaking updates, product information, technical white papers, press coverage, and customer success stories.

Sun Directory Services Administration Guide (http://docs.sun.com:80/ab2/coll.297.1/@Ab2CollToc?subject=sysadmin)—Describes the Sun Directory Services.

Netscape Directory Services documentation (http://home.netscape.com/eng/server/directory/)—Describes the Netscape Directory Services.

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P-2 Shell Prompts

Shell	**Prompt**
C shell prompt	`machine_name%`
C shell superuser prompt	`machine_name#`
Bourne shell and Korn shell prompt	`$`
Bourne shell and Korn shell superuser prompt	`#`

SIMS User Registration

Register as a user of the Sun Internet Mail Server 4.0 (SIMS) to receive information about new releases, upgrade offers, and promotions. To register, press the *Registration* button at the Administration Console login page. Fill-in the form requesting your name, address, e-mail address, and other information, and press *Send*. When Sun receives the completed registration form, we will email an acknowledgment back to you. You must provide an email address in order to receive a confirmation notice.

Error Conditions

Registration errors are rare, but TABLE P-3describes the possible error messages and the required action.

TABLE P-3 Registration Error Conditions and Required Action

Error Message	**Required Action**
"The server could not set a locale to encode the mail. There was no locale supplied and the server could not set the default properly."	Make sure that you start registration from the Admin console screen.
"The server could not obtain the <LOCALE> locale that you registered from to properly format the mail. It is necessary to have the same locale installed on the server that you registered from."	Either make sure the locale installed on server is the same as the locale you are registering from on your client or, type in registration in us-ascii.
"The mail program on the server could not be opened."	There was an error involving the sendmail program. Make sure that /usr/lib/sendmail is on your system and properly configured.
"There was not enough memory to process the mail."	You've run out of swap space. Shut down applications or increase swap and try again.

Notice

To better illustrate the process being discussed, SIMS manuals contain examples of data that might be used in daily business operations. The examples might include names of individuals, companies, brands, and products. SIMS manuals use only fictitious names, and any similarity to the names of individuals, companies, brands, and products used by any business enterprise is purely coincidental.

CHAPTER 1

Introduction

Service providers are discovering that customers want them to provide a wide array of electronic services including, email services. To meet these needs, Sun™ has recently released Sun™ Internet Mail Server™ 4.0.

Topics in this chapter include:

- What is Internet messaging?
- What is Sun Internet Mail Server?
- What is new in Sun Internet Mail Server 4.0?

What is Internet Messaging?

The Internet has effectively lowered the cost of electronic communication. As the number of people and organizations connected to the Internet has grown, the Internet has evolved into a new channel for communication. To facilitate Internet services, Internet messaging clients and easy-to-use web browsers have provided cost-effective way of publishing and sharing information with employees inside the enterprise as well as customers, suppliers, and partners outside.

Messaging services has become crucial to enterprise infrastructure in the 1990s. Organizations are seeking messaging solutions that provide a lower cost of ownership while increasing the effectiveness and reliability of their communications network. Specifically, they are evaluating the benefits of Internet standards-based messaging systems.

Standard-based messaging systems typically include native or gateway support, some of which include:

- **Simple Mail Transfer Protocol (SMTP)**—for reliable, end-to-end transfer of messages
- **Multi-Purpose Internet Mail Extensions (MIM**E)—for transmitting rich or complex messages
- **Post Office Protocol version 3 (POP3)**—for delivering messages to millions of POP3 capable clients
- **Internet Message Access Protocol version 4 (IMAP4)**—for state-of-the-art message interoperability between clients and servers
- **Lightweight Directory Access Protocol (LDAP)**—for creating an integrated directory service to centralize user management
- **Standard Network Management Protocol (SNMP)**—for enabling network wide management of the messaging system
- **Secure Sockets Layer (SSL)**—for providing robust, enterprise-ready encryption

What is Sun Internet Mail Server?

Sun Internet Mail Server is a full-featured, Internet standards-based that enables reliable, encrypted electronic messaging inside and outside the enterprise. It supports all the standard-based messaging systems, including SMTP, MIME, POP3, IMAP, LDAP, SNMP, SSL, DNS, UUCP, and TCP/IP, to create a single, low-cost messaging infrastructure for internal and external communications.

Sun Internet Mail Server 4.0 enables service providers to extend their messaging service infrastructure from basic consumer hosting services to messaging hosting services for corporate customers.

Architected from the ground up, SIMS scales from workgroups to thousands of concurrently active users.

With native support for virtual domains and delegated management, Sun Internet Mail Server 4.0 increases the flexibility of deployment for service providers as they extend their messaging services.

SIMS Key Features

Sun Internet Mail Server is based entirely on end-to-end, native Internet Standards.

The key features of Sun Internet Mail Server include:

- **IMAP4 server**—enabling centralized message storage and disconnected support
- **POP3 server**—supporting popular POP3 e-mail clients
- **Message Store**—providing safe, high-performance, scalable repository of mail messages
- **Multi-threaded MTA**—providing scalability and high message throughput
- **LDAP directory**—providing centralized network user, resource, and distribution information
- **Java**^{TM} **software**—providing administration for all administration features
- **On-line documentation**—enabling easy access information for the product
- **SSL 3.0 client/server security**—providing reliable messaging services

---

Note – See "SIMS Components Features" in Chapter 5, "SIMS Architecture" for a complete list of the features that the SIMS 4.0 product offers.

SIMS 4.0 Benefits

Key benefits of Sun Internet Mail Server 4.0 include:

- **Lower total cost of ownership**—approximately one-third the initial hardware/software price of Microsoft Exchange
- **Performance**—as many as 120,000 concurrent IMAP4/POP3 users on single server
- **Scalability**—three to four times the numbers of users on equivalent hardware compared to other mail systems
- **High reliability**—based on the Solaris operating environment
- **High availability**—based on Sun Enterprise Cluster 2.2 architecture
- **Various Sun clients**—including Sun Web Access, MAPI, and Solaris DTMail
- **Wide range of standards**—based on third-party clients tested with top twelve Internet mail clients

What is New in Sun Internet Mail Server 4.0?

The key features of Sun Internet Mail Server 4.0 includes virtual domain hosting and delegated management capabilities that enable the service providers to deliver outsourcing services to their corporate consumers.

Sun Internet Mail Server 4.0 is specifically designed for service providers who are moving from hosting traditional consumers messaging to corporate message application hosting (business outsourcing). It provides a secure and reliable platform on which corporate customers could host multiple messaging services.

New SIMS 4.0 Key Features

The features new to Sun^{TM} Internet Mail Server^{TM} 4.0 include:

- **Virtual hosted domains suppor**t—Service providers can host messaging for several companies on one server, enabling corporate customers to preserve their unique corporate identities.
- **Web-based Delegated management console**—Web HTML console for SP's customers to perform day-to-day management tasks, such as user and distribution list management, resulting in lower management costs for SPs and increased flexibility for their customers.
- **Extensive Provisioning tools**—Customer are allowed to provision against the directory for both batch applications and integration with billing and management interfaces.
- **Enhanced Directory Server support**—The product supports Netscape Directory Server 4.1 as well as Sun Directory Server 3.1.
- **Service Level restrictions**—Allows customer to set controls on access (POP, IMAP, HTML) and quotas on a domain and user basis.

CHAPTER 2

SIMS Key Features

Many businesses have realized that the ability to allocate their email services by outside vendors is cost and time effective to their businesses. Hosting a large number of email services for these organizations is then accomplished by service providers.

This chapter describes how the key features of the Sun™ Internet Mail Server™ enable service providers to deliver corporate outsourcing.

Topics in this chapter include:

- Virtual hosting support
- Vertical and horizontal scalability support
- Delegated management capability
- Provisioning interfaces
- Monitoring tools

Virtual Hosting

Corporations demand that their Internet addresses look and feel as if they own their dedicated server, even when they are being hosted by an service provider (SP). Key to this concept is the Internet DNS namespace. For example, on the public Internet, email must be addressed to `smith@stream.com` and not to `stream@bridge.net`. Similarly, on the private Intranet, users connect their mail clients and browsers to `stream.com` and not some magic name inside `bridge.net`. Besides reinforcing the subscriber's own identity, this also separates them from changes in their SP.

Historically, SPs have approached this by dedicating their entire servers to each subscriber domain, which is too expensive for very small accounts. To address this need, SIMS enables the individual servers to support many domains, giving the impression that each domain has its own server. This capability is known as *virtual hosting*. In turn, the concept of the DNS domains hosted by such servers is known as *hosted domains*.

SIMS provides an automated process for creating and managing new virtual hosts in the LDAP directory, which creates appropriate rewriting rules in the IMTA.

See "Address Rewriting Rules" in Chapter 7, "Internet Messaging Transfer Agent" for more information on rewrite rules.

See Chapter 3, "Domain Hosting with SIMS" for domain hosting components and specifications.

See "Virtual Hosting Scenario" in Chapter 4, "Deployment Scenarios" for a case study of a virtual hosting scenario.

Vertical and Horizontal Scaleability

SIMS 4.0 supports populating a large number of users in the directory. The key criteria are search speed from the message access servers, the rate of addition of new entries, the rate of modification of existing entries, the time to synchronize the IMTA (incremental and full), and the time to synchronize LDAP Slave servers with the master. Achieving these goals require improved caching, smart installation tools, application specific tuning tools, and enhancements to the directory server itself.

To implement these requirements, SIMS provides both vertical and horizontal scalability models. *Vertical scalability* refers to maximizing the use of the available hardware, adding new resources to enable an existing server run proportionately faster. *Horizontal scalability* refers to the ability to connect multiple servers so that they act as a single logical server. The performance of the logical server can be increased by adding more physical servers.

See “Horizontal Scalability Scenario” in Chapter 4, “Deployment Scenarios” for a case study of an horizontal scaleability scenario.

Delegated Management Capability

SIMS 4.0 Delegated Management capability enables an SP who provides email services to a customer to outsource the administration of that customer’s mail domain to the customer. This ensures that customers can only perform a prescribed set of operations on a prescribed set of entries and attributes residing only in that part of the directory which corresponds to the customer’s mail domain.

To perform these administration tasks, SIMS allows creating one or more *delegated administrators* for each hosted domain. A delegated administrator is able to create and edit users and distribution lists within the specific domains. A single level of delegation is provided; that is, the delegated administrator is unable to delegate sub domain managers. SIMS provides delegated management capability by using any Web browser. The GUI look and feel for the Delegated Management Console is customizeable.

Provisioning Interfaces

SIMS provides provisioning interfaces to enable the SPs write applications for integrating their order entry, billing, and systems management software with the mail server.

SIMS also publishes the Directory and IMTA configuration interfaces to facilitate the migration tools between the Mail server and customer account databases, like Oracle or Sybase. This includes the ability to create, modify, and delete new virtual hosts.

See the *Sun Internet Mail Server 4.0 Provisioning Guide* for information on provisioning the SIMS LDAP directory with users, distribution lists, administrators, and domains by creating and importing LDIF records.

Anti-spamming Services

The common term *anti-spamming* refers to a broad set of services that protect the mail server and its user community from the undesirable effects of Unsolicited Commercial E-Mail (UCE), popularly referred to as *spam*. The first priority is to protect the server from denial of service attacks, which have become increasingly common. The second priority is to reduce the clutter in subscriber’s mail folders, where UCE can cause lost messages due to exceeding quota limits.

Monitoring Tools

SP sites deploy many tools to assist in site status monitoring, error recovery, tracking down security violations, billing customers for usage, and other administrative tasks. Many use commercial enterprise monitoring systems, where as others use entirely home grown tools. These tools require that the mail server provides published interfaces for monitoring server health, obtaining historical data (logs and audit trails), gathering statistics, and administering the system.

Two types of monitoring interfaces are provided:

- **Control and Monitoring**—Internal control CLIs and dynamic monitoring parameters, supporting integration of systems management software (for example, Tivoli) with the mail server.
- **Domain Logging and Statistics**—Provide detailed per-domain logging and report generation capabilities, allowing writing applications to generate their own report formats and billing systems.

CHAPTER 3

SIMS Topology

This chapter shows the Sun™ Internet Mail Server™ enterprise and services provider messaging configurations.

Topics covered in this chapter include:

- Enterprise messaging topology
- Enterprise messaging topology with workgroup domains
- Service provider messaging topology
- SIMS as a Proxy mail access server

Enterprise Messaging Topology

This section shows two ways that SIMS can be configured for Bridge's internal email system at `bridge.com`. Notice that this view is different from the view of the email system offered to Bridge's SP customers at `bridge.net`.

SIMS Enterprise Configuration

FIGURE 3-1 shows a configuration with two SIMS systems set up at Bridge Corporation that are located behind a single firewall. This configuration supports two geographically separated offices.

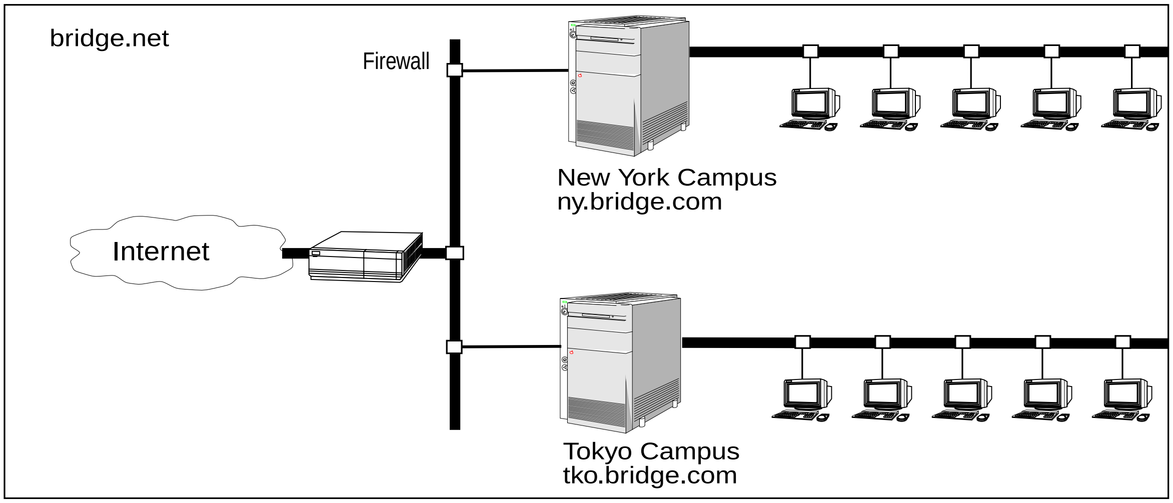


FIGURE 3-1 SIMS Enterprise Messaging Topology

In this configuration, incoming mail comes to one of the SIMS servers where it is stored in a user's. Outgoing mail goes to the server which, in turn, will either send it to a local mailbox, to one of the other SIMS servers in the intranet, or relay the message out to the Internet.

Enterprise Messaging Topology with Workgroup Domains

FIGURE 3-2 depicts a more complex environment with three work groups at the New York campus of Bridge.

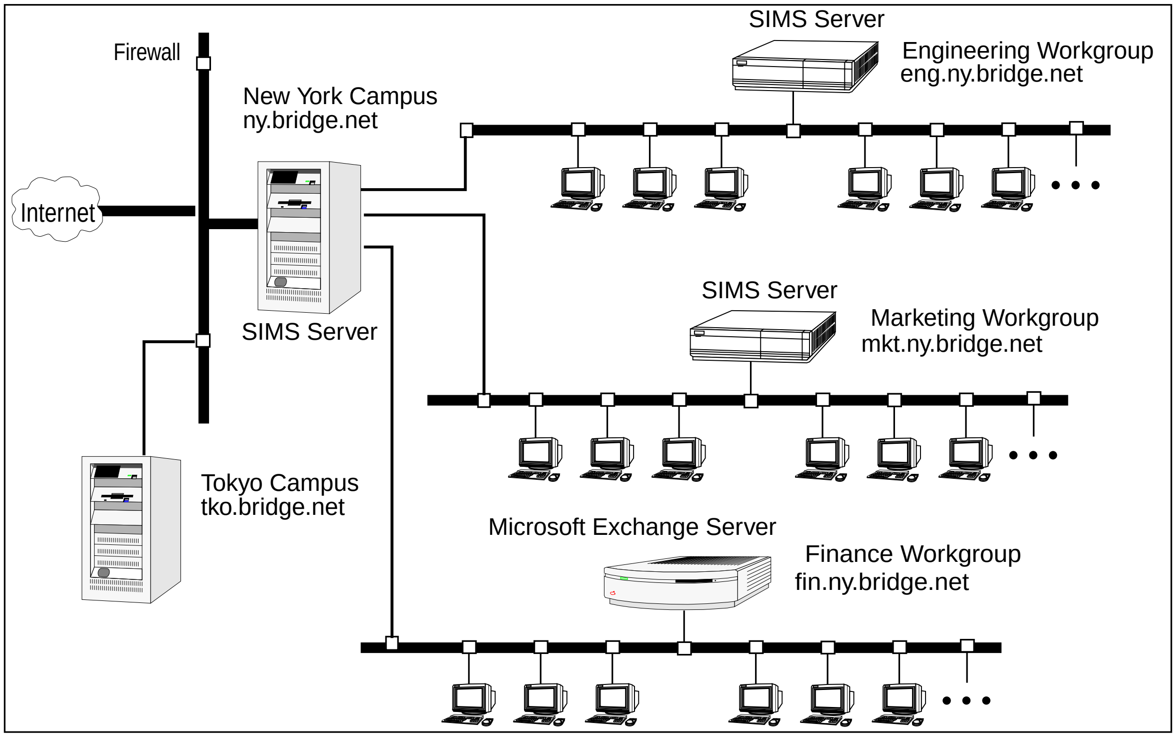


FIGURE 3-2 SIMS Enterprise Messaging Topology with Workgroups Domains

In this topology, each of the three separate groups within Bridge has its own domain that is being hosted by a local mail server. The engineering domain and the marketing domain run SIMS, while the finance domain runs Microsoft Exchange. The main New York server also runs SIMS and is responsible for routing incoming mail to the appropriate workgroup server. Each of the workgroup servers stores messages and makes them available to the local mail clients. Outgoing mail is sent from the client to the local domain server to the New York campus server. The campus server relays the message to the appropriate Internet server outside the firewall or to the Intranet server inside the firewall.

The Tokyo campus server uses SIMS to support the email system in Japan.

Services Provider Messaging Topology

Services providers have somewhat different needs and requirements than enterprises. Generally they serve a much larger customer base, they are more oriented toward POP than IMAP, and they prefer downloading email user messages to their customer's machines rather than store them in-house.

The SIMS environment in FIGURE 3-3 provides email services for Bridge's customers. It uses multiple SIMS systems that are configured to perform different responsibilities.

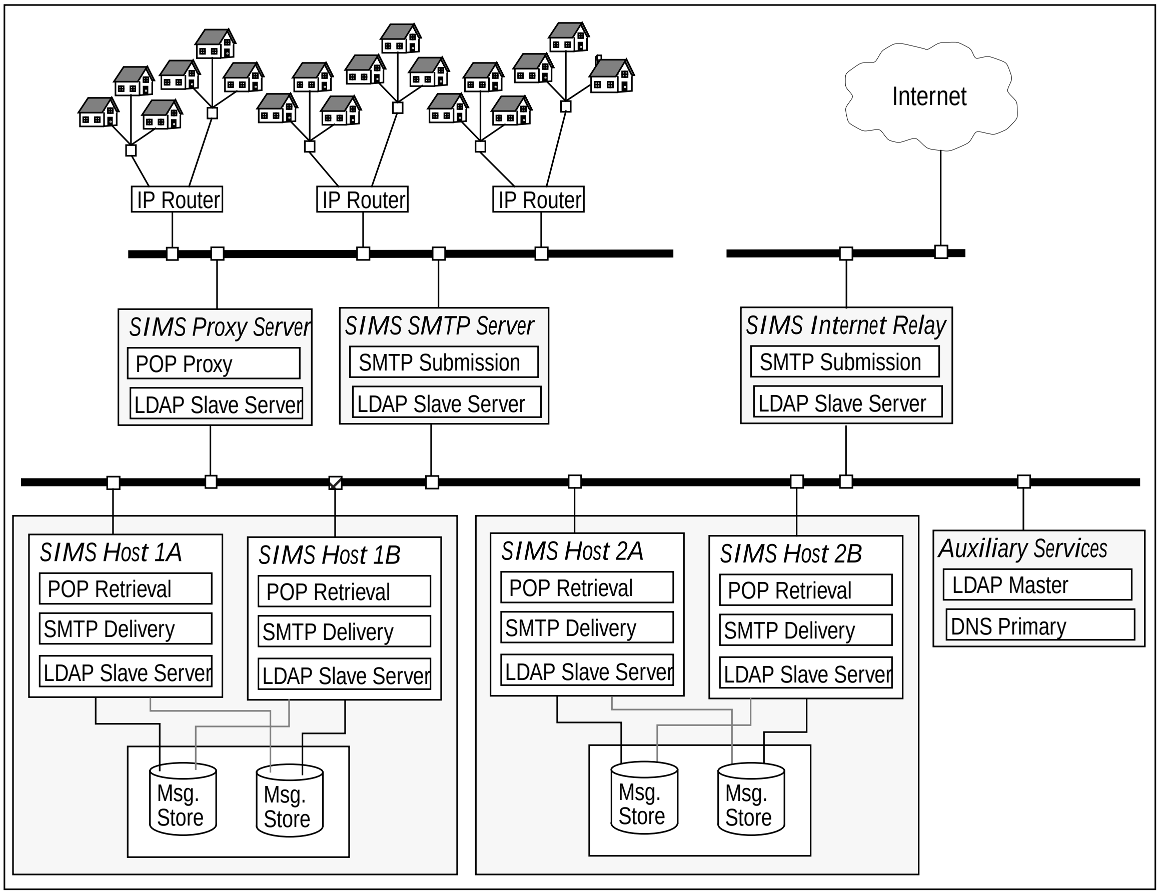


FIGURE 3-3 Service Provider Messaging Topology

This system depicts a large SP using eight copies of SIMS configured to handle different responsibilities. The houses represent the SP's customers. The two large shaded boxes at the bottom represent two large message stores. Two hosts for each message store exist since the SP has installed the SIMS High Availability option. If one of the message store hosts fails, for example SIMS Host 1A, the system will fail over to the backup SIMS Host 1B.

When a customer wants to retrieve their mail, the POP request (most SPs support POP although SIMS can support IMAP if so configured) goes through the SIMS Proxy server which first authenticates the user and then forwards the request to Host1 or Host2, depending on the location of the customer's mailbox. Mail is then downloaded to the customer's machine. See "SIMS as a Proxy Mail Access Server."

A message being sent by a customer goes through the SIMS SMTP server. The SMTP server determines whether the message destination is local or over the Internet. If the message is local, however, it sends the message via SMTP to the IMTA at either SIMS Host1 or SIMS Host2. The IMTA then sends the message to the local message delivery tool, `ims_master`. If the message is not local, the message is sent to the *SIMS Internet Relay* machine where it is forwarded to the Internet.

Mail coming from the Internet is received by the SIMS Internet Relay. If the message is local, it is sent to the message store at SIMS Host 1 or 2. If not local, the message is relayed to the next stop in the Internet.

Note that in this configuration the SIMS Proxy server, SIMS SMTP server, and the SIMS Internet Relay do not support a message store of any type.

In this deployment, a separate machine is set up to run as the LDAP master. Master directory information is maintained here, and changes to the master are filtered down to each of the SIMS hosts which acts as an LDAP slave. The auxiliary services also act as the primary DNS server.

See Chapter 5, "SIMS Architecture" for overviews of components, features, and system view of SIMS.

See Chapter 4, "Deployment Scenarios" for descriptions of several detailed scenarios, representing different models with which SIMS could be outsourced.

SIMS as a Proxy Mail Access Server

A SIMS proxy mail message access server (or simply *proxy*) operates like a regular SIMS server, but does not support a local message store. Instead, it receives POP/IMAP mailbox access requests, and then forwards these requests to the SIMS system containing the requested mailbox. FIGURE 3-4 depicts a proxy setup that shows the POP/IMAP mail retrieval path.

Proxies enable horizontal scalability (the ability to transparently expand the capacity of a SIMS environment by adding more SIMS servers) and Internet access to private Intranet mail systems. See Appendix A, "Configuring SIMS as a Proxy Message Access Server" in the *Sun Internet Mail Server 4.0 Administrator's Guide* for detailed information on configuring proxies.

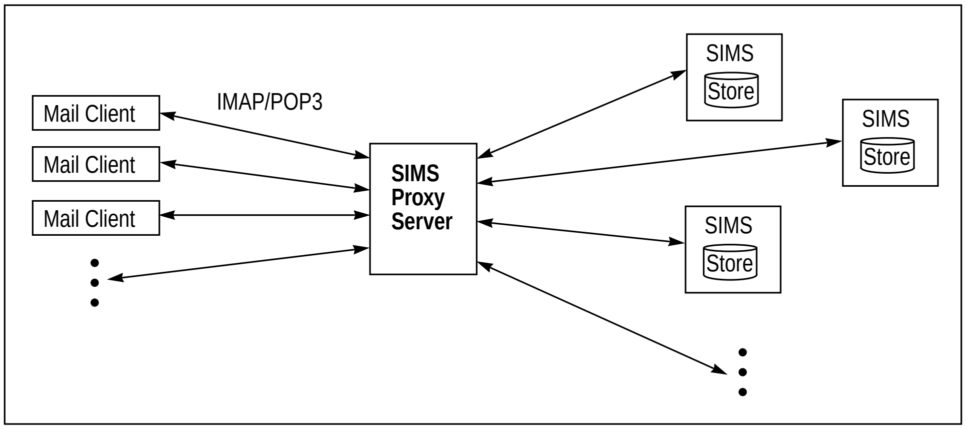


FIGURE 3-4 SIMS as a Proxy for Internet Access

CHAPTER 4

Deployment Scenarios

This chapter provides a case study from which three different scenarios—consumers, virtual hosting, and horizontal scalability—are prepared.

SIMS concepts will be described within the context of a *common usage model*: a case study of a company deploying SIMS as its email system. By describing the SIMS concepts around the various scenarios that come out of the case study, this chapter should provide a context for these concepts.

Topics in this chapter include:

- Defining a case study
- Consumer hosting scenario
- Virtual hosting scenario
- Horizontal scalability virtual hosting scenario

Defining a Case Study

The first step for defining a case study is to define the requirements and configurations of the subscribers that are to be hosted by a service provider. This section identifies these criteria.

See Chapter 6, "Domain Hosting with SIMS" for domain hosting feature components and specifications, including the Delegated Management Console and Domain Management server.

Protocol Requirements

The scenarios in this case study assume that only native Internet clients are being supported, using the standard Internet email protocols (SMTP, RFC822, MIME, IMAP, POP3, and so on.).

Bridge Corporation Characteristics

This case study uses an SP called Bridge Corporation: a rapidly expanding SP (telecommunication and Internet) company. Bridge uses two domains: `bridge.com` for its internal mail and `bridge.net` for its customers.

Bridge's Services

The following are the types of services that the example SP, Bridge Corporation, will provide for its subscribers:

- Email hosting
- Web site hosting
- Delegated Management hosting
- Electronic commerce hosting
- Mail applications

Subscribers Characteristics

This case study uses four different email subscribers:

stream.com

Summary:	Medical group.

Users: Approximately 150 employees.

Email Usage: Ownership and control of electronic distribution of company approved news bulletins and specialized mail distribution lists.

green.gov

Summary: Environmental organization with interests in extensive research.

Users: 20,000 spread among three permanent offices in the U.S., Mexico, and South Africa. 5,000 who work primarily in the field, at research and consumer sites.

Email Usage: Email is a primary means of corporate communication, including a variety of departmental distribution lists.

forest.edu

Summary: Regional community college.

Users: 100 full time staff and faculty, 75 to 100 interim and visiting faculty. 3,000 to 6,000 students, all at a single site.

Email Usage: Administration uses email to distribute class schedule and other announcements. Faculty use at their discretion. Email accounts offered to all students while they are actively enrolled.

ocean.org

Summary: Non-profit scientific research organization.

Users: 30 employees at a single site.

Email Usage: Internet savvy. It uses email to communicate with other organizations interested in their research, and to issue newsletters to contributors.

Directory Service

In these scenarios, all of the user and domain properties are stored in a Directory Service. Objects are retrieved from the Directory through the Internet Standard Lightweight Directory Access Protocol (LDAP).

Directory Information Tree (DIT) Structure

The Directory Information Tree (DIT) is organized to provide a simple algorithmic mapping between *distinguished names* and *Internet domain names*. See Chapter 8, “Directory Services” for definitions on distinguished names.

The DIT for this case study is shown in FIGURE 4-1.

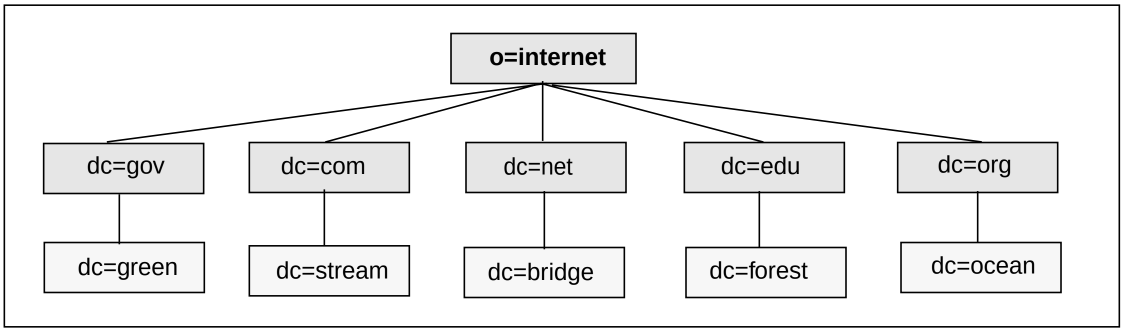


FIGURE 4-1 Directory Information Tree for the Bridge SP Case Study

Domain Distinguished Names

The Directory Component (DC) tree is rooted at the name `O=Internet.` Each of the components of the domain name is mapped to a domain component relative distinguished name. TABLE 4-1 shows the mapping of the subscriber names that are used in this case study

TABLE 4-1 Domain Distinguished Names (DN)

Domain Name	Distinguished Name
bridge.net	dc=bridge;dc=net;o=internet
stream.com	dc=stream;dc=com;o=internet
green.gov	dc=green;dc=gov;o=internet
forest.edu	dc=forest;dc=edu;o=internet
ocean.org	dc=ocean;dc=org;o=internet

Consumer Hosting Scenario

Traditionally, the SPs have provided access, such as email and other Internet-based services to the consumers or residential.

See Chapter 3, "SIMS Topology" for Enterprise and SP messaging topology for the SIMS system.

Consumer Hosting Topology

FIGURE 4-2 shows the configuration for the consumer hosting of email services.

On the Consumer's Site

Historically, the SP provides the consumer with just a plain, unprotected IP router. The consumer operates and maintains their own application servers, including the email server, DNS server, and (if needed) LDAP server. For their own protection, the consumer must operate through a firewall that filters out undesirable packets and insulates the organization's internal network from the Internet. Notice that for many organizations, especially small ones, the email server may actually be the firewall system.

FIGURE 4-2 shows only a single mail server per consumer. Organizations larger than 100 members or more usually deploy separate hosts for their direct connection to the Internet (the MX-based host) and for their mailboxes.

Recall that each mapping from a domain name to an IP address is called a record in DNS. Among several types of records is the *MX record*, which is short for Mail Exchange. To process an email message, a mail server program such as `sendmail` looks for this record of the destination address before checking other types of records. An example of an MX record in the master DNS database for the domain that is to receive the email is *MX 10 mailhost*. The DNS database, in turn, is a file that contains mappings from the name of a domain or a host to an IP address. The host which keeps this database is called the DNS server.

This ensures that if the MX-based host is attacked from the outside, users within the organization will be able to read their mail and exchange new email with each other. Many sites will use multiple MX-based hosts, for reliability if not for improved bandwidth. And even modestly sized companies will use multiple internal mail servers, as fits their own internal organization.

Consumer to Service Provider Connection

A consumer site connects to the SP Point of Presence through either *leased line* or *dialup*. In *leased line* access, the consumer has a permanent physical connection between their network and the SP. This provides the most reliable service, and is required if the consumer is hosting interactive services like HTTP, FTP, or Remote Access. Popular mechanisms include analog leased line, Digital Direct Data service, T1, Fractional T1, and T3. Newer mechanisms that are currently growing in popularity include cable modems, DSL, and ATM.

In *Dialup* access, however, there is no connection between the consumer site and the SP except when there is traffic. For very small sites or organizations that have very little Internet traffic, this can be much cheaper than a leased line. The most common mechanisms are analog dialup over standard telephone lines and ISDN.

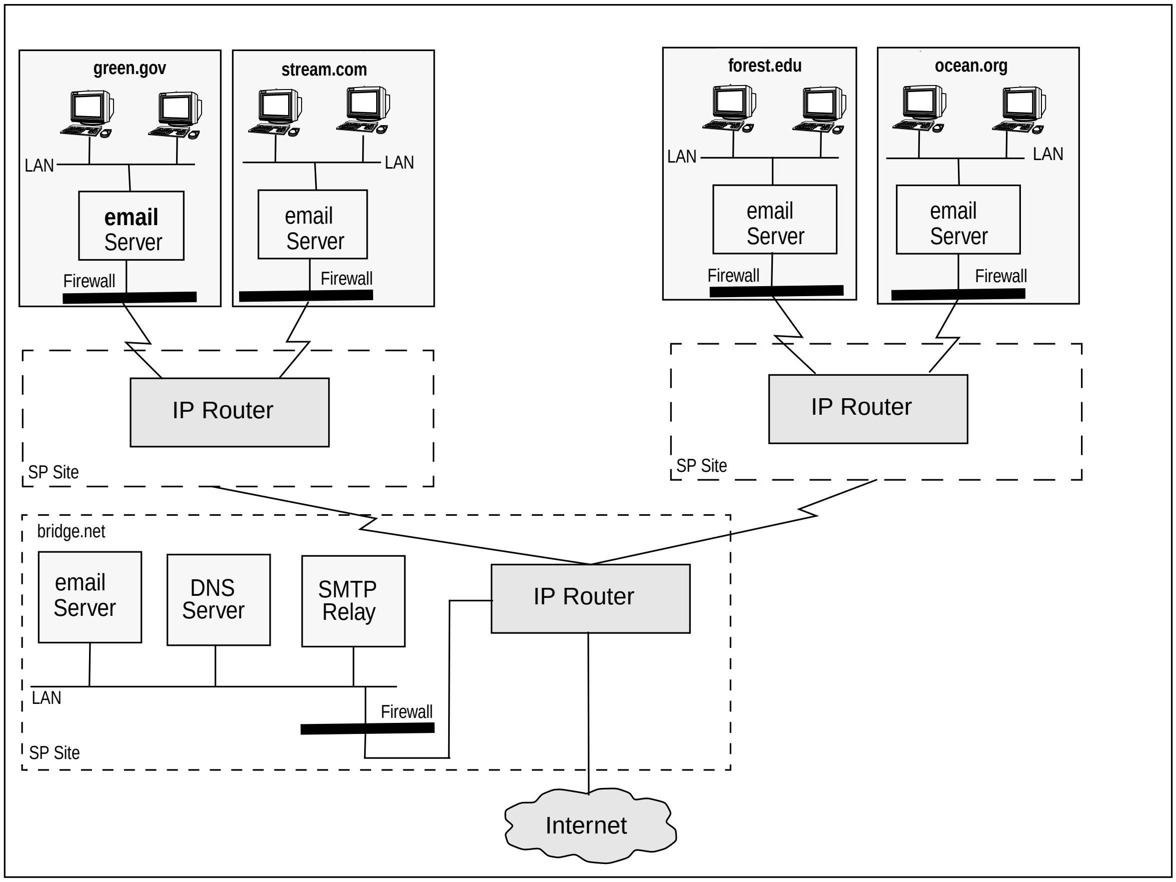


FIGURE 4-2 Consumer Hosting Scenario

Service Provider Email Server

This email server is associated with mail access systems. The SP consumers who wish to read their mail will need to connect to this server. IMAP/POP proxy could be configured here.

DNS Server

The SP operates a DNS server that provides the following services:

- Primary master server for the SP's own domain, `bridge.net`
- Designate as the root server for all consumers
- Primary master server for consumers who do not wish to maintain their own public DNS server
- Secondary server for consumers who prefer to maintain their own public server.

SMTP Relay Host

The consumer hosting scenario shows an SMTP *relay* host that is managed by the SP. This can offer a number of value added services, for which the SP may charge additional fees.

The relay host can be configured as a low precedence MX host for the consumer's domains. This allows the relay host to accept and hold the consumer's email when their mail server is down.

Certain low volume consumers use dialup access, as explained above. The relay host provides message storage for these consumers when they are not connected. These sites must periodically poll the relay host to retrieve their incoming email.

The relay host imposes a significant management burden on the SP. Consumer email may live on this server for an indefinite time, raising issues of backup and failure recovery. If one of the consumer servers fails because of being swamped, then the consumer's email may roll over to the SP's relay host. Because of this, most SPs do not offer a relay host for those consumers that are hosting their own email server.

Directory Service

In the consumer hosting scenario, the LDAP Directory server is located at the consumer's site, which can be operated by the consumer. Currently, most organizations do not expose their LDAP servers to the public network for security reasons.

Virtual Hosting Scenario

Consumer hosting required a separate mail server to be supported by the SP for each domain. While this architecture is well understood and easy to manage, it is not cost effective for small domains. In addition, as the number of domains increases, the management of the individual services becomes increasingly unwieldy. For this reason, virtual hosting that supports a single mail server which can support many domains would be the choice.

The goal of virtual hosting is to provide an environment in which the difference between physical versus virtual hosting is transparent to the consumers.

Virtual Hosting Topology

FIGURE 4-3 shows the architecture for the virtual hosting topology. The difference between the consumer hosting scenario and the virtual hosting scenario is that the virtual scenario uses fewer number of servers to service the domains for which it is hosting.

On the Consumer's Site

The resources used at the consumer site are the same as used at a consumer hosting scenario, where the consumer's mail servers are moved to the SP's site. Ideally, the consumer should not need to make any changes to the configuration. This may not be achievable, however, because of the limitations in IMAP, POP3, and SMTP.

Consumer to Service Provider Connection

Most business consumers who purchase email outsourcing will use the same type of connections to connect to the SP site.

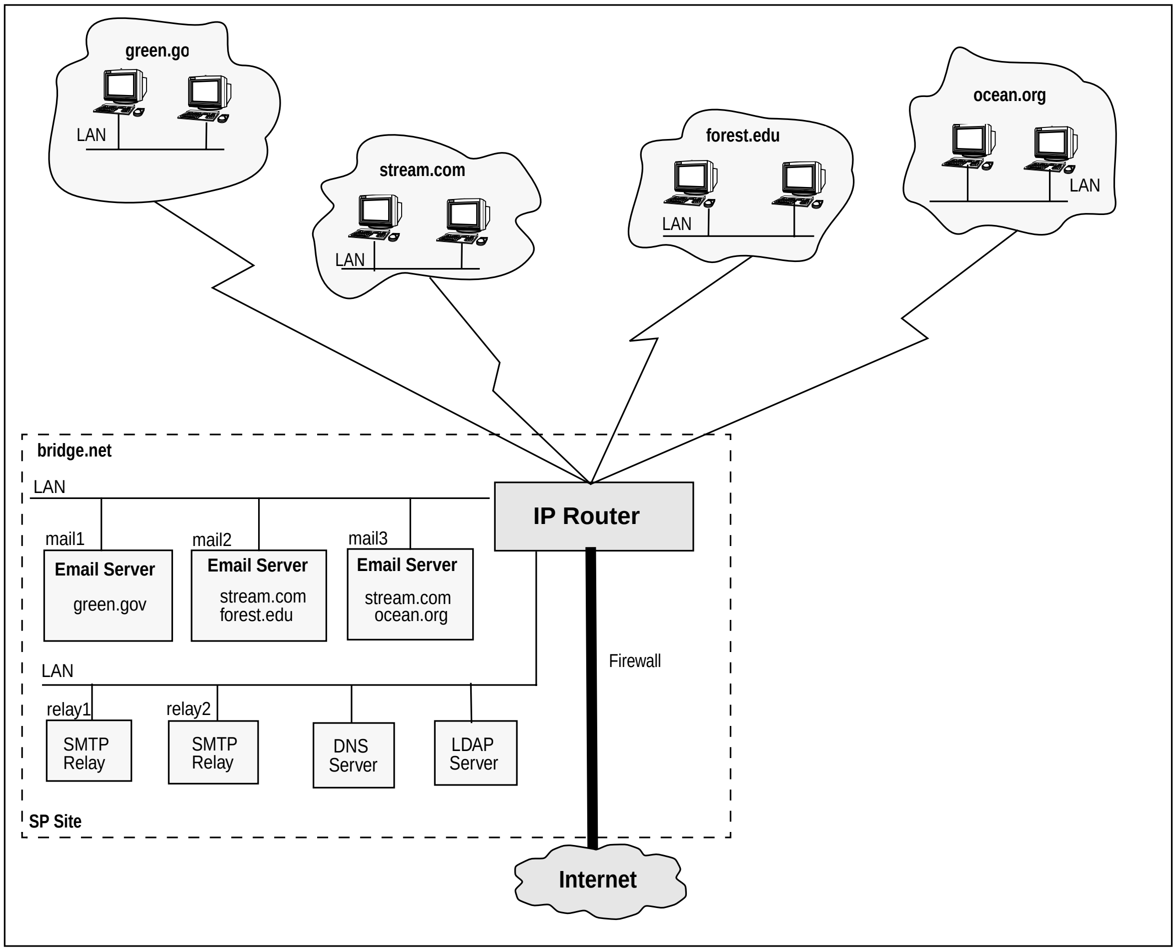


FIGURE 4-3 Virtual Hosting Scenario

Email Servers

These servers provide the interface to the consumers, including POP3 and IMAP message storage and retrieval, SMTP message submission, and (if offered) HTML-based mail client services.

As shown in FIGURE 4-3, `green.gov` is assigned to the first mail server, `stream.com` and `forest.edu` is assigned to the second mail server, and `stream.com` (again) and `ocean.org` and are assigned to the third mail server.

In a classical Internet email server, the domain name is discarded when the messages are delivered into the message store. This behavior is still supported, and is called the *Canonical Domain* for the server. In addition, the server supports any number of hosted domains, where the server preserves the domain name when the messages are delivered into the message store.

An SP will normally set the canonical domain to its own domain name, for example, `bridge.net`. If this server is being used for both residential users and outsourcing users, then it may makes sense to assign the home users to the canonical domain, and the outsourcing users to an appropriate hosted domain. If the server has only outsourcing users, then the canonical domain may be empty or used only for administrative mailboxes.

SMTP Relay Hosts

The two SMTP relay hosts provide the mail interface to non-consumers. Their function is essentially identical to that used in the consumer hosting.

All incoming email is directed to these hosts via MX records, and is then forwarded to the specific server via the routing tables in the IMTA. All outgoing email is directed to these hosts via the routing tables in the consumer email servers.

Directory Service

The LDAP directory is used by the IMTA to retrieve local user and group address information. When the IMTA receives a message, it uses the directory information to determine where the message should be delivered. The message store uses the directory services to authenticate users logging into their mailboxes. The message store also obtains information about user message quota limits and message store type (IMAP or POP).

Horizontal Scalability Scenario

In general allocating a server in a virtual hosting scenario that is capable of supporting the entire user base is not practical. Distributing different domains over different servers does distribute the load, but it requires that all users for a domain be contained on the same server.

Adding horizontal scalability to virtual hosting provides the following benefits:

- Domains can span over more than one email server, either to improve availability or because the domain has too many users to be supported by a single server.
- New users can be added to whichever server has the lowest load. A new server can be brought online and immediately start supporting new users from all domains, without having to move existing users to the new server.
- Authentication and Web Access services (HTML to email) can be off-loaded from the backend email servers.

Horizontal Scalability Topology

FIGURE 4-4 shows the architecture for horizontally scalability scenario. There are two fundamental changes from the consumer hosting scenario. First, each mail server is now responsible for more than one single consumer domain, just as was the case with the virtual hosting that was described in the previous chapter. Second, instead of connecting to general purpose email servers, the consumers connect to Proxy servers, which hide the underlying distribution of consumers across multiple backend message store servers.

On the Consumer’s Site

The resources used at the consumer site are the same as used in a consumer hosting scenario, where the consumer’s mail servers are moved to the SP’s site. Ideally, the consumer should not need to make any changes to the configuration. This may not be achievable, however, because of limitations in IMAP, POP3, and SMTP.

Consumer to Service Provider Connection

Most business consumers who purchase email outsourcing will use the same type of connections to connect to the SP site.

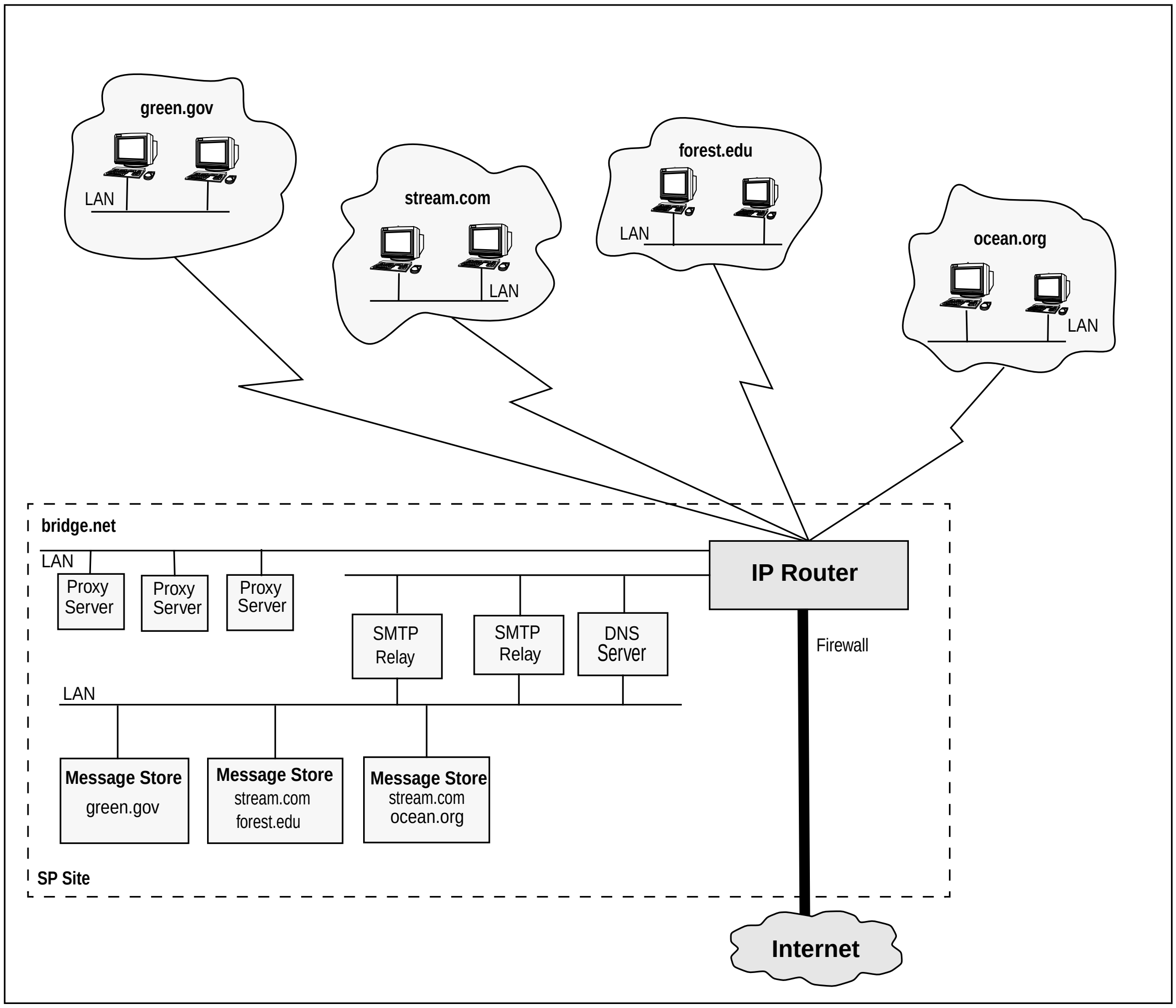


FIGURE 4-4 Horizontal Scalability Scenario

Message Proxy Servers

These servers provide the interface to the consumers, including POP3 and IMAP message retrieval, SMTP message submission, and (if offered) HTML-based mail client services. The proxy servers insulate the end users from the underlying mail server architecture, as well as off loading some computation from the message stores servers.

The POP3 and IMAP services are true proxy servers. For each connection, they authenticate the user, consult the Directory Services to determine which message store server contains the consumers INBOX and folders, open a connection to the message store server, and then drop into a *pass through* mode, relaying packets between the client and the real server.

The SMTP service functions as a relay. Messages for other SP consumers are routed to the appropriate message store server; however, messages for users outside the SP are routed to one of the relay hosts.

Note that the Proxy servers are identical, except for their node name and IP addresses. The only persistent data are messages in the IMTA queues. If needed, servers may be added to support the load.

Message Store Servers

The message store servers hold each user's inbox and personal folders. The Proxy servers and the SMTP relay hosts are the only hosts that connect directly to the message store servers.

SMTP Relay Hosts

The two SMTP relay hosts provide the mail interface to non-consumers. All incoming email is directed to these hosts through MX records. All outgoing email is directed to these hosts via the routing tables in the Proxy servers. Their function is very similar to that used in the consumer hosting.

CHAPTER 5

SIMS Architecture

Sun™ Internet Mail Server™ is an extensible framework of cooperative modules that create an enterprise-wide and open standards-based messaging system. SIMS core components consists of the transfer agents, message store, and access units, and directory services.

Topics in this chapter include:

- SIMS components overview
- Message routing through SIMS
- SIMS messaging data flow example
- SIMS 4.0 system view
- Supported standards
- Supported clients

SIMS Components Overview

FIGURE 5-1 shows the components that comprise SIMS architecture. These components are:

- Internet Message Transfer Agent
- Message Store / Message Access
- Directory Services
- Sun Web Access Server
- Delegated Management Console
- Delegated Management Server
- SIMS Administration Server

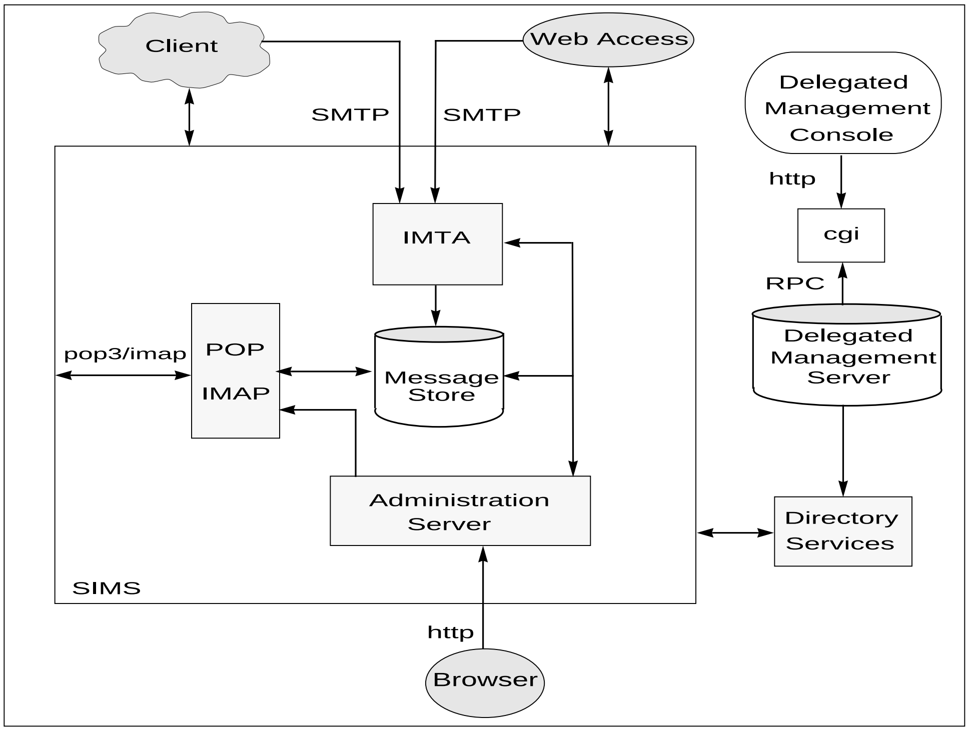


FIGURE 5-1 SIMS Components

See "Overview of SIMS Components" for overviews of each components of SIMS.

SIMS Optional Components

The following are SIMS 4.0 optional components that you may choose to *add* to your SIMS core server.

Additionally, you could install these options as *stand-alone* components on servers where SIMS is not installed.

- Message Transfer Agent SDK
- SIMS 4.0 Documentation Set
- Remote Administration Console

SIMS Components Features

This section shows the features that the SIMS 4.0 product offers based on its individual components. Items that are identified as *New* are specific to the 4.0 release.

Internet Message Transfer Agent

- Powerful anti-spam configuration with anti-relaying
- SMTP authentication *New*
- POP before SMTP authentication *New*
- Scalable channel architecture
- Domain hosting support *New*
- DNS canonicalization to qualify non-FQDN names and normalize hostname aliases *New*
- SDK for custom application development
- Extensive address rewriting (including address reversal) and channel management facilities
- Pipe channels supporting the IMTA through native Solaris programs

Message Store/Message Access

- Domain hosting support *New*
- Domain/subscriber authorized services through POP and IMAP *New*
- APOP authentication for POP3 *New*
- POP before SMTP connection *New*
- Concurrent access to any message folder *New*
- Secure Socket Layer (SSL) messages access
- Message store utilities *New*
- Integrated backup/restore utilities

Sun Web Access Server

- Domain hosting support *New*
- Domain and user provision *New*
- HTML-based
- Brandable
- Single integrated UI to view email, directory, and calendar
- Message attachment support
- Sun Web server support *New*
- High Availability support *New*

Delegated Management Console

- Domain-level user creation and management *New*
- Domain-level distribution list and management *New*
- End-user personal preferences setup *New*
- Customizable and brandable *New*
- HTML-based application *New*

SIMS Administration Console

- Netscape browser support *New*
- Domain creation and deletion *New*
- Domain service restrictions set up *New*
- Domain-level user creation and management *New*
- Distribution list setup and management *New*
- Server statistics
- Server components configuration

- Server monitoring and queue status
- Starting and stopping server
- Purging deleted entries

Administration Utilities

- Domain creation and management *New*
- Domain and user authorized service restrictions *New*
- Domain-level user creation and management *New*
- Delegated administrator creation *New*
- Bulk-loading new domain-level users *New*
- Changing user authorized services *New*
- Distribution list setup and management *New*
- Service and performance monitoring *New*

Directory Services

- Netscape Directory Services 4.1 support *New*
- Sun Directory Services 3.1 support *New*
- Multiprotocol, distributed, scalable, client/server-based global directory
- LDAP v3 server *New*
- Server failure rollover *New*
- LDAP address referral in master/slave configuration *New*
- Remote LDAP sever support *New*
- Remote user authentication *New*
- Domain Component (DC) tree structure support *New*

Other Optional Features

- Message Transfer Agent SDK
- SIMS 4.0 Documentation Set
- Remote Administration Console

Overview of SIMS 4.0 Components

This section provides overviews of the SIMS components, as listed below:

- Internet Message Transfer Agent
- Message Store/Message Access
- POP3/IMAP4 protocol
- Directory Services
- Sun Web Access Server
- Delegated Management Console
- Delegated Management Server
- SIMS Administration Console
- SIMS Administration Utilities
- SIMS Server Man Pages
- SIMS High Availability systems
- Remote Administration Console
- Message Transfer Agent SDK

Internet Message Transfer Agent

The Internet Message Transfer Agent (IMTA) routes, transports, and delivers Internet Mail (RFC 822) messages within the email system.

The IMTA performs all of its operations on a set of *channels*. The two types of channels are internal and external. An *internal channel* is an interface between the internal modules of the IMTA. Internal channels include the reprocessing, conversion, and defragmentation channels. These channels are not configurable from the SIMS Administration Console.

An *external channel* is an interface between the IMTA and another SIMS component, such as the Sun Message Store, or another component outside of SIMS (for example, the Internet or a local mail client). The external channels are configurable.

See Chapter 7, “Internet Message Transfer Agent” for detailed information on the IMTA component.

Message Store/Message Access

Sun Message Store is a dedicated data store for the delivery, retrieval, and manipulation of Internet email messages. This message store works with the IMAP4 and POP3 servers that are integrated with SIMS. It saves any message that conforms to RFC 822 specifications, and recognizes the Multipurpose Internet Mail Extensions (MIME) content format.

The advantage of the Sun Message Store is the ability to save only a single copy of any incoming message that is sent to a distribution list or multiple recipients, provided all recipients are on the same mail server. For example, if a message is sent to 20,000 users on the same server, only one master copy of the source message is saved in the store.

POP3/IMAP4 Protocol

Post Office Protocol Version 3 (POP3) is an implementation of the server side of the POP3 (RFC 1939) access protocol standard. Many popular mail clients currently use POP3.

Internet Mail Access Protocol Version 4 is an implementation of the server side of the standard IMAP4 protocol (RFC 2060). IMAP4 is used by client mail applications to access Internet email messages in distributed, enterprise/Internet-wide message stores. Messages are parsed on delivery to ensure the highest IMAP performance.

See Chapter 9, “Sun Message Store” for detailed information on the message store component.

Directory Services

SIMS 4.0 supports both the Netscape Directory Services as well as the Sun Directory Services. Integrated with SIMS is Sun Directory Services 3.1 that provides a multiprotocol, distributed, scalable, client/server-based global directory. It allows storing information such as user definitions, user profiles, network resource definitions and configuration parameters. It supports a range of naming, directory, and authentication protocols on the top of a shared and distributed repository.

Sun Directory Services 3.1 is ready for use with leading Web browsers, PC address book tools, and client software. It is fully compatible with other mail and directory applications.

See Chapter 8, “Directory Services” for information on the available directory services and components, and directory replication.

Sun Web Access Server

Sun Web Access is a client application that gives end-users browser-based access to the SIMS 4.0 email and Name Directory services and to the Solaris Calendar server. Sun Web Access is integrated with the SIMS 4.0 server system and centrally administered. See the *Sun Internet Mail Server 4.0 Web Access Administrator's Guide* for information on configuring and administering this server.

Delegated Management Console

Delegated Management Console is an enhancement to SIMS. It enables a service provider, a reseller of Internet services who provides email services for their customers (subscribers), to delegate the administration of that customer's mail domain to the customer. This delegated management can perform on a prescribed set of users and groups within a prescribed set of operations of the customer's mail domain.

While SIMS provides all the email facilities for a hosted domain, the Delegated Management Console enables the SP's subscribers to create and manage domain-level users and distribution lists as well as setting up end-user personal preferences within the user's hosted domain.

See Chapter 6, "Domain Hosting with SIMS" for domain hosting feature components and specifications.

See Chapter 11, "Delegated Management Administration," for summaries of tools that different types of administrators could use to perform domain hosting administrative tasks.

See the *Sun Internet Mail Server 4.0 Delegated Management Guide* for descriptions of the Delegated Management Console and the tasks associated with the console.

Delegated Management Server

The Delegated Management server provides directory services to the Delegated Management Console. Once it has interpreted a request from the Delegated Management Console, the server performs the necessary access controls checks. If the access controls are positive, it then applies the changes to the Directory Services and relays the directory's response back to the Delegated Management Console. If the access controls are negative, the server denies the access to the Delegated Management Console.

SIMS Administration Console

The SIMS Administration Console is a GUI-based administration tool that the SIMS administrator can use to setup, maintain, configure, and monitor the SIMS system, including the domain hosting capabilities.

The console allows configuring SIMS components such as Message Transfer Agent (IMTA), Sun Messages Store (MS), Message Access (MA), Directory Services, and monitoring and queue status.

The SIMS administrator at the SP site can use the console to create hosted domains as sub-domains within its own domain as well as peer domains. The SIMS administrator is enabled to delegate these tasks to the delegated administrator without compromising the integrity or security of the server.

Alternatively, the SIMS administrator could perform user and distribution list administrative tasks for hosted domains by using the SIMS Administration Console at the SP site.

See Chapter 10, “SIMS Administration Console” to learn about the SIMS Administration Console services and components.

See Chapter 11, “Delegated Management Administration” for summaries of the tasks and tools that are available to the delegated administrators.

See Chapter 11, “Configuration Files” in the The *Sun Internet Mail Server 4.0 Installation Guide* for lists of files that are associated with each SIMS component.

See the *Sun Internet Mail Server 4.0 Administration Guide* for instructions on how to configure, maintain, monitor, and troubleshoot your mail server using the SIMS Administration Console.

SIMS 4.0 Administration Utilities

The SIMS 4.0 Administration Command Line Interface (CLIs) provides administrators with a set of command line utilities to manage a SIMS system. The current methods of managing a SIMS system include manually editing configuration files, running various component specific scripts or executable, and running the SIMS Administration Console.

These methods require intimate knowledge of the underlying components as well as their interdependencies and is prone to error. However, managing a large number of users or automating repetitive tasks is not feasible through the SIMS Administration Console. The CLIs will provide another method of managing a SIMS system which will alleviate these problems.

Like the SIMS Administration Console, the SIMS administrator at the SP site can use the CLIs to create and manage hosted domains, as well as to delegate an administrator to administer users and distribution lists within that hosted domain.

Both the SIMS administrator and the delegated administrator could use the CLIs to create users and distribution lists within a hosted domain, provided that they both have UNIX accounts.

See the Administration CLI man pages for descriptions of command-line utilities and detailed information about the arguments and attributes relevant to each command.

SIMS Server man pages

UNIX manual reference pages for the SIMS server command-line utilities and configuration files. The man pages provide detailed information about the arguments and attributes relevant to each utility. The configuration file man pages provide detail about the file's structure and parameters.

SIMS High Availability System

SIMS 4.0 is a high-performance, highly scalable mail delivery and access system. Since mail is critical to enterprises and SP customers, many SIMS customers wish to run SIMS on a cluster to get higher availability of their mail system through fail-over to a surviving member of the cluster.

For this reason, SIMS 4.0 provides *Asymmetric High Availability* (HA) configurations. In this configuration, all the SIMS binaries, configuration files, message queues, and message stores reside on a shared disk. When a switch-over occurs, the disk is unmounted from the failing system and mounted on the surviving system.

See the *Sun Internet Mail Server 4.0 Installation Guide* for instructions to install and configure SIMS/HA, and for descriptions of the architecture of this system.

Remote SIMS Administration Console

The Remote SIMS Administration Console allows you to remotely access and use the SIMS Administration Console on any other Solaris, Windows 95, and NT systems that is not running SIMS. It allows you to administer the SIMS server from a remote server.

Message Transfer Agent SDK

The Message Transfer Agent SDK is a set of C APIs for the Internet Message Transfer Agent (IMTA). It allows you to develop custom SMTP channels, such as a custom channel to send mail to a user’s pager or to a fax machine.

Message Routing Through SIMS

Submitted messages from the Internet or local clients go to the IMTA via SMTP (Simple Mail Transport Protocol). If the message address is within the SIMS domain, the IMTA delivers the message to the message store. If the message is addressed to another domain, the IMTA relays the message to another transport agent on the Internet or Intranet.

Messages to the local domain are stored in the message store or the traditional UNIX `/var/mail` file system depending on how the system is configured. Once messages are delivered to the appropriate mailbox, they can be retrieved, searched for, and manipulated by IMAP4 or POP3-based mail clients.

The IMTA uses the LDAP directory to retrieve local user and group address information. When the IMTA receives a message, it uses the directory information to determine where the message should be delivered. The message store uses the Directory services to authenticate users logging into their mailboxes. The message store also obtains information about user message quota limits and Message store type (IMAP or POP). Outgoing client messages go to the SMTP channel in the IMTA. The IMTA sends the message to an Internet IMTA or, if the address is local, to the message store.

SIMS Messaging Data Flow Example

FIGURE 5-2 shows the routing of messages through the SIMS system.

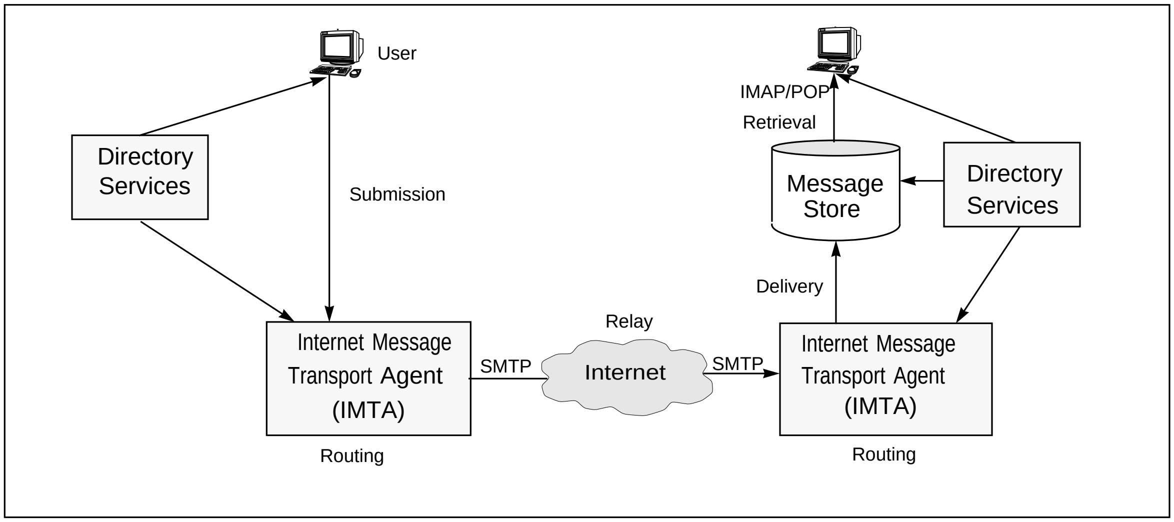


FIGURE 5-2 SIMS Message Routing

The message data flow is as follows:

1. The user submits a message to be delivered to another user on the network. The mail client sends the message to the IMTA via the SMTP protocol.
2. The IMTA reads the address and the routing information from the directory service server. It determines delivery information from the address, consulting the IMTA directory cache and domain rewriting rules as appropriate.
3. Having the routing information, the IMTA sends the message to the IMTA on the receiving end.
4. This IMTA reads the address and looks up the host and mailbox information in the directory services server for the receiving client. If the message address is within the SIMS domain, the IMTA delivers the message to the message store. If the message is addressed to another domain, the IMTA relays the message to another transport agent on the Internet or Intranet.
5. The receiving client logs in to the message store with the user’s password. The message store verifies that the password corresponds with the username and then allows access to the user’s messages.
6. The user can now retrieve the message or delete it.

SIMS System View

FIGURE 5-3 shows a SIMS system that contains all SIMS components and the relationship between these components. It also shows the connection between this SIMS server and the clients that SIMS supports.

See “Supported Clients” for a list of clients that SIMS supports.

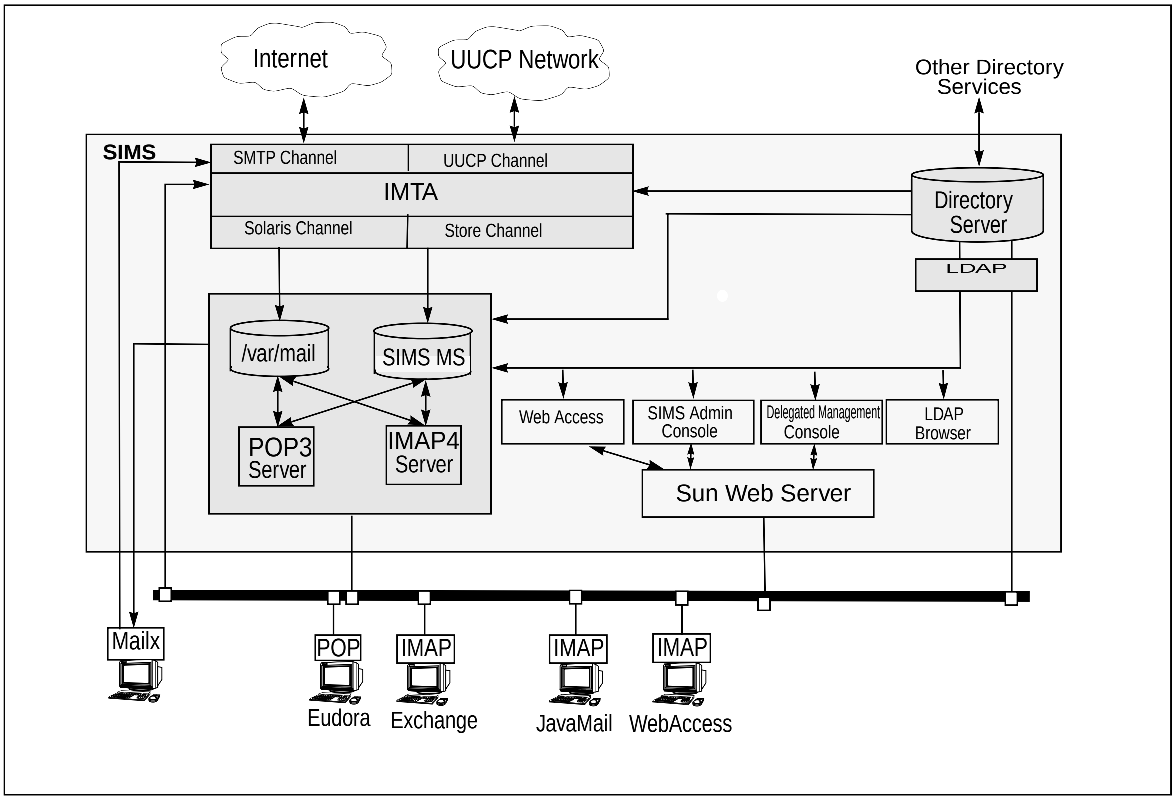


FIGURE 5-3 SIMS System View

Supported Standards

TABLE 5-1 shows the standards that the SIMS IMTA and message store and Access components use. The table also identifies the types of message store that SIMS supports.

TABLE 5-1 Supported Standards

Component	Protocol
Internet Message Transfer Agent (IMTA)	Transport protocols: ESMPT, MIME, UUCP
	Value added channels: FAX, Pager, Printer (future)
Message Store and Access (MS/MA)	Message Access Protocols: IMAP, POP3
	Message Stores: Sun Message Store, `/var/mail`

Supported Clients

Sun Internet Mail Server supports a wide range of clients. It features a single message store for POP3, IMAP4, and OpenWindows Mailtool environments, which enables you to have a single mail server for PC, UNIX, and Macintosh environments.

Sun Internet Mail Server is tested with some of the most popular Internet mail clients such as Netscape Messenger, Microsoft Outlook Express, Qualcomm's Eudora Pro, CommTouch Software's Pronto E-Mail, and NetManage's Z-Mail.

If you have existing investment in non-Internet standards-based clients such as Microsoft Windows Messaging Inbox or Microsoft Outlook, Sun Internet Mail Server software delivers tools to protect your investment. SIMS 4.0 contains MAPI service providers for IMAP and LDAP that give these clients access to SIMS.

Sun Internet Mail Server 4.0 software also features Sun Web Access, which enables accessing server-based email, calendar, and directory information by any standard Web browser. It enables users to stay in touch anytime, anywhere.

CHAPTER 6

Domain Hosting with SIMS

Service providers expect to be able to host a large number of email services for different organizations. These businesses have recognized that the ability to allocate their email services to the SPs would save time and cost for their organizations.

In observing the need of the SPs, SIMS 4.0 provides email services that enable multiple organizations to have their own virtual domain.

This chapter describes the SIMS domain hosting feature components and specifications.

Topics in this chapter include:

- What is domain hosting?
- Domain hosting components
- Domain hosting authentication
- Domain hosting restrictions

What is Domain Hosting?

The service provider Bridge Corporation provides email services to the Stream company, which does not wish to manage its own Internet domain, `stream.com` (although it owns the domain name). Stream is dependent on `bridge.net` to supply email services to the `stream.com` domain. Although the mail server for all email users in the `stream.com` domain is <*mailhost*>`.bridge.net`, a user sends and receives mail as <*user*>`@stream.com` instead. That is, `bridge.net` does not appear in their email addresses.

SIMS 4.0 supports the ability to define such domains in the directory and host them on a shared mail server. This concept is referred to as *domain hosting*. It is also referred to as virtual hosting in Chapter 4, "Deployment Scenarios." FIGURE 6-1 shows a typical domain hosting configuration. Notice that the message stores are presented logically.

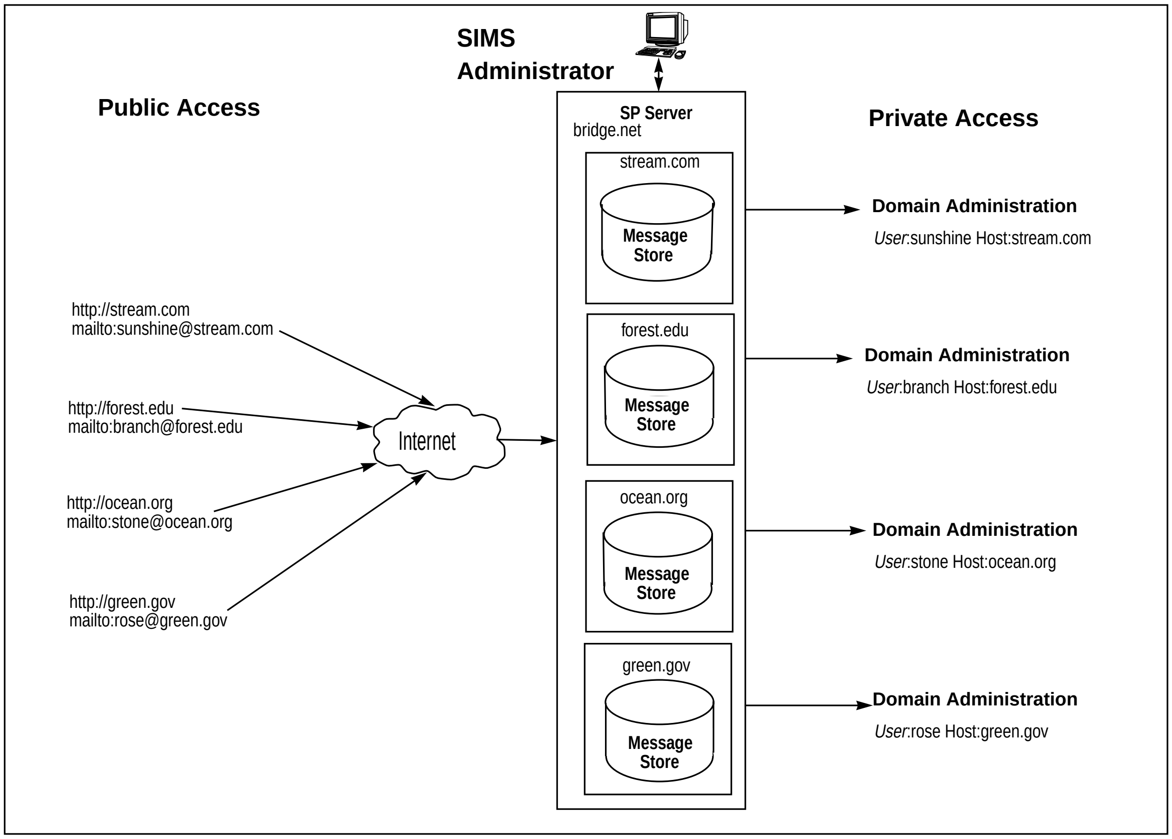


FIGURE 6-1 Domain Hosting Configuration

Domain Hosting Components

SIMS domain hosting capabilities comprise the following components:

- Delegated Management Console
- Domain Management server
- Delegated administrator
- LDAP directory

See Chapter 4, “Deployment Scenarios” for detailed information on the case study used in the examples in this chapter.

See Chapter 5, “SIMS Architecture” for an overall picture of all components that comprise SIMS architecture, including domain hosting components.

See Chapter 11, “Delegated Management Administration” for summaries of tools that different types of administrators could use to perform domain hosting administrative tasks.

Delegated Management Console

The Delegated Management Console is a set of HTML forms and CGI programs that delegated administrators may use to perform management tasks. The delegated administrator submits requests from its browser through HTML forms. The Delegated Management Console component translates the delegated administrator’s request into the Delegated Management server. The Delegated Management server returns values to the Delegated Management Console component, which, in turn, interprets server’s responses and dynamically generates HTML forms from them.

See the *Sun Internet Mail Server Delegated Management Guide* for more information on the Delegated Management Console.

Delegated Management Server

The Delegated Management server is an RPC server that provides directory services to the Delegated Management Console. Once it has interpreted a request from the console, it performs the necessary access control checking. If the access controls are positive, it then relays the request to the Directory Services and relays the directory’s response back to the console. If the access controls are negative, it generates a negative response to the Delegated Management Console.

Delegated Administrator

The delegated administrator is the administrator at the hosted domain site who is dedicated to managing these hosted domains.

See Chapter 11, “Delegated Management Administration” for summaries of tools that different types of administrators could use to perform domain hosting administration tasks.

See the *Sun Internet Mail Server Delegated Management Guide* for more information on different tasks that the delegated administrator can perform using the Delegated Management Console.

LDAP Directory

LDAP Directory is the master repository of all the information related to hosted domains.

None of the other SIMS components store permanent hosted domain information. That is, the message access server retrieves the necessary information to associate a client with a domain. Similarly, the IMTA retrieves hosted domain information to perform proper routing and address rewriting.

Domain Hosting Authentication

Each user must have a unique user ID within a hosted domain that they will use within the mail system. Although externally, user `smith` at `Stream` would receive his email at `smith@stream.com`, he would log in to POP3/IMAP4 using his user ID, `smith`, and domain name, `stream.com`. This method of logging in is very typical of how SPs set up the mail systems.

These user IDs are mapped to and from the `user@host.com` entries by the IMTA from the information stored in the directory.

Internal Unique User ID

All of SIMS components rely on unique user names that are used by the message store as the mailbox directory name and by the IMTA as keys in its alias database.

SIMS 4.0 uses the LDAP user attribute uid as the unique name. Besides being a unique key, `uid` also defines the IMAP/POP login name. Although this works in an environment where uid is guaranteed to be unique within the entire mail server, it is not the case with hosted domains, where uniqueness of the mail user names is enforced on a per-domain basis (one can have the same `uid smith` in both `stream.com` and in `ocean.com`), and a single server can host several domains.

To address this need, SIMS 4.0 uses a combination of the uid and the DNS domain name of the user to guarantee uniqueness.

Domain Hosting Restrictions

Transmission of Access Rights

A delegated administrator is not able to grant administrator-level rights to another user in its domain. It has to be done by the SIMS administrator at the SP site.

Control of Access Rights

SIMS 4.0 requires that all delegated administrators have the same rights. It is not possible for the SIMS administrator to give different permissions to delegated administrators. The SIMS administrator can, however, assign multiple delegated administrators to perform different tasks. For example, some delegated administrators could be responsible for changing passwords while others can change user information. All delegated administrators will be able to make changes to all of these fields.

Browsing Capabilities for Users

End-users can not browse the directory through the delegated admin interface.

CHAPTER 7

Internet Message Transfer Agent

The Internet Message Transfer Agent (IMTA) routes, transports, and delivers Internet mail messages for SIMS.

Topics in this chapter include:

- Overview of IMTA
- IMTA structure
- Email message structure
- IMTA directory cache
- Address rewrite rules
- SMTP Authentication
- Controlling SMTP email access
- Distribution lists

Overview of IMTA

The Internet Mail Transport Agent (IMTA) is a general-purpose, store-and-forward system for distributing computer-based mail. The term *store-and-forward* means that the IMTA automatically handles the receiving of mail messages necessitated when network links or services are temporarily unavailable. In contrast to mail user agents (MUAs) such as Netscape Messenger or Microsoft Outlook, which are used to create and read electronic mail messages, IMTA is a mail transport agent (IMTA) responsible for directing messages to the appropriate network transport and ensuring reliable delivery over that transport.

The IMTA provides a uniform distribution environment that can be interfaced to multiple mail user agents (MUAs), networks, protocols, and transport mechanisms. As this interfacing, from the user's point of view, is accomplished transparently, the IMTA presents to the user a homogeneous mail network; that is, the IMTA blends heterogeneous mail networks into a single, coherent mail system.

IMTA Structure

One analogy for the IMTA is that it performs for email messages what a central transportation transfer station performs in a city. That is, passengers in a city may come in to a station by using different way of transportations—by foot, road, subway, railroad, by air. Over some of these transports there may be alternate protocol possibilities, for instance, taxi cabs, buses, and cars travel over roads. Another example is that commercial airlines and private airplanes provide variant forms of air travel. Depending on each passenger's destination, the passenger departs by way of an appropriate transport and protocol to get him to his next destination.

In the context of electronic mail, protocols are generally a high-level (not necessarily network specific) language spoken between two mailers. Transports are the low-level, network specific details used to implement a protocol on a given network.

Thus email messages may come in to the IMTA by any one of a variety of transports and protocols—submitted directly by a local user, via TCP/IP as an SMTP message from an Internet system, by using a dial-up modem using the PhoneNet protocol, DECnet as a MAIL-11 message, DECnet as an SMTP message, UUCP, an X.400 transport, SNA, and so on. The IMTA then routes the message out using a transport and protocol appropriate for the message's destination address.

Note that the IMTA not only serves as a mechanism for sending and receiving mail, but also serves as a centralized switching yard or *gateway* for routing and rerouting mail traffic between multiple network transports. The use of the IMTA as a mail gateway allows the IMTA host to provide electronic mail access through its network facilities for other, less capable machines.

The Transportation Methods—IMTA Channels

The IMTA uses what are called *channels* to implement specific combinations of transports and protocols. Each different transport and protocol combination has an associated IMTA channel. The IMTA postmaster initially configures the IMTA telling it what sorts of transports and protocols are in use at his site, and what sorts of destination addresses should be routed through which sorts of channels. For instance, at sites with an Internet connection, Internet addresses are normally routed through an SMTP over TCP/IP channel; but at sites with only a UUCP connection, Internet addresses would instead be routed through a UUCP channel. Once the IMTA is so configured, the IMTA handles message routing and delivery automatically—ordinary users need never be aware of this underlying transport and routing; that is, they simply address and send their messages and the IMTA automatically routes and delivers them appropriately.

The Layout of Transportation Routes—The IMTA Configuration File

The IMTA's main configuration file, the `imta.cnf` file, contains the fundamental IMTA configuration information for a site, the information about what sorts of transports and protocols are in use (the IMTA channel definitions), and the information about which destination addresses should be routed through which channels (the IMTA rewrite rules). The IMTA configuration file is discussed in the IMTA Configuration chapter of the *Sun Internet Mail Server 4.0 Reference Manual*.

The IMTA configuration file thus contains a site's *layout* in terms of transports and protocols and which destinations are reachable via what transports: akin to the physical layout of railroad tracks, bus lines, airline routes, and so on for a transportation transfer station in a city.

Arrivals Trigger Activity—IMTA Channels Run on Demand

There is also the issue of scheduling: when do messages arrive and when are they delivered (when do the trains, buses, airplanes, and so on arrive and depart)?

For incoming messages, in most cases, the underlying transport is simply configured to hand the messages over to the IMTA immediately whenever they come in. The IMTA is *passive* or *data driven*, waiting for the messages, which may come in at any time. For some protocol and transport combinations, the component of the IMTA that awaits incoming messages for that protocol and transport combination is implemented as a server; for instance, to listen for and receive incoming SMTP over TCP/IP messages, the IMTA includes an SMTP server. In other cases, for some transports where the transport itself is not able to actively hand the messages over to the IMTA, the IMTA is configured to periodically *poll* the sending side and ask for incoming messages.

Since the IMTA is data driven, when an external source submits a message into the IMTA, the receiving IMTA channel processes the message to check where the message is destined and then hands the message over to the appropriate outgoing channel and triggers the outgoing channel to itself run in turn. (In a message with multiple recipients, the receiving channel hands the message over to multiple outgoing channels and triggers each of the outgoing channels to run.) In particular, note that for outgoing messages, by default the IMTA normally always makes an immediate attempt to deliver each message.

To recapitulate, the physical transportation transfer station analogy above in the IMTA the normal scheduling of outgoing message deliveries is *on-demand* and immediate—it is rather as if each new passenger could get their own railcar or airplane with immediate departure.

Delivering the Passengers—The Execution of IMTA Channel Processing Jobs

The execution of the incoming direction of an IMTA channel may occur in a user’s own process (as for messages submitted by local users on the IMTA system), as an IMTA server process running (as for incoming SMTP messages), or as an IMTA channel job triggered in some other way. The receiving channel immediately triggers the execution of a subsequent channel job to perform the next step of delivery of the message, and these subsequent jobs triggered by the IMTA itself run in the IMTA processing queues. The IMTA processing queues are controlled by the IMTA Job Controller, which has its own configuration options. Configuration of exactly how and when channels run in processing queues can be used to control characteristics of the IMTA operation.

The Waiting Room—The IMTA Queue Area on Disk

While messages are awaiting processing and delivery by the IMTA, they are stored as message files on disk in the IMTA queue area for the destination channel.

Delivery Retry Attempts

The IMTA is a store-and-forward message system. If the IMTA's attempt to deliver a message encounters a temporary failure condition, then the IMTA stores the message and later attempts another delivery. There are IMTA periodic jobs that run every so often re-attempting delivery of not yet delivered messages. Eventually, if a message still cannot be delivered after repeated attempts, the IMTA periodically returns (bounces) the message back to the sender. How long the IMTA keeps trying to deliver messages is of course configurable.

Types of Operations

IMTA performs five different operations:

- **Layout**—The IMTA configuration describes how the IMTA should handle messages and addresses and message content.
- **Receiving Messages**—IMTA servers or incoming channel jobs receive messages. In general, the IMTA is data driven. The IMTA waits, listening for external agents to submit messages whenever they wish.
- **Processing Outbound Messages**—The IMTA channel jobs running in the IMTA processing queues deliver outbound messages. Note that a job in an IMTA processing queue may be either an immediate delivery job, or a periodic job re-attempting delivery of not-yet-delivered messages.
- **Scheduling of Delivery Retries and Polling**—For messages that do not get delivered upon first attempt, a periodic delivery job periodically attempts another delivery. Note that periodic jobs are a backup to the normal immediate delivery attempts.
- **Message Storage**—The IMTA queue area is where message files are stored while being processed and awaiting delivery.

  In addition to the major issues of message routing and delivery, sites often wish to modify the addresses in e-mail messages, or convert message contents. Some simple address modifications are configured simply as part of the basic email routing in the IMTA configuration file. For more complex needs, the IMTA has extensive address handling facilities for address aliasing, mailing lists, centralized naming, and so on.

Email Message Structure

Most IMTA messages are stored as text files. Messages with multiple parts (possibly containing different types of data) are represented as a series of text sections separated by special unique delimiter strings.

Briefly, the key items in each message file are:

- The message envelope. The first records in the file contains message envelope (that is, transport) information.
- The envelope is terminated either by a line containing a boundary marker, or by a line containing two CTRL/A characters.
- The header lines of the message follow the envelope. Their format is mandated by RFC 822.
- There may be any number of message header lines; the message header formed by this collection of header lines is terminated by a single blank line after which follows the message body.
- The message is terminated by a boundary marker matching the boundary marker at the beginning of the message, (or by a sequence of five CTRL/As if the message start was indicated using CTRL/As).
- For messages that had transient delivery failures information about retrial would go here.

Note – Sun reserves the right to change this format in future releases of the IMTA. User written applications that either read or write queued IMTA message files should make use of appropriate IMTA library routines. Use of the IMTA SDK will insulate user applications from any future message format changes. For complete information on message envelopes and headers, respectively, refer to RFCs 821 and 822.

Message Envelope

The IMTA uses the contents of the envelope to make routing decisions. It does not use the content of the message. The content of the envelope is primarily defined by RFC 821. It includes the originator address, the recipient(s) address(es), and envelope ID. The IMTA supports additional envelope information related to SMTP service extensions published after RFC 821 such as notary (the ability to specify requested Delivery Status Notification for each recipient, see RFC 1891) and original recipient addresses.

Message Headers and Contents

RFC 822 defines a message as headers and contents.

Message Headers

An Internet mail message starts with one or more headers. Each header is composed of a field name followed by a colon then a value. Values can be generated by the composer of a message, the mail client, and MTAs. Headers contain the following types of information about the message:

- Delivery information (for example, TO, CC, BCC, From, reply-to, in-reply-to)
- Summaries of the content (for example, subject, keywords, comments).
- Information that enables tracing of a message if problems occur (for example, message-ID, references, received, return-path).
- Information specific to the Multipurpose Internet Mail Extensions (MIME). For more information on MIME and MIME-specific headers, refer to Message Content/MIME.

RFC 822 defines several headers. Not all defined headers need to be present in a message. In fact, only a few headers are required for any type of message.

The administrator of a mail client can construct a template of desired headers that the composer of a new, forwarded, or replied-to message fills in. The following is an example of a simple template of headers for a new message:

To:
Subject:
Date:
Cc:

The mail client can automatically generate some headers (From, reply-to, message-ID, and references).

Before the message is submitted to the IMTA, the mail client can add a date header. If the From header contains multiple email addresses or if the email address is different than the submitting mail client, then the sender header is added.

An IMTA can also add headers to a message. Each IMTA that accepts a message adds a received header to that message. The last IMTA to accept the message and to actually deliver the message to the message store adds a return-path header. The received and return-path headers provides information that enables you to trace the routing path taken by the message if a problem occurs.

Message Content / MIME

A blank line separates the headers and the content or body of the message. The content or body of the message provides the data that the originator of the message intends to transmit to the recipient.

SIMS supports Multipurpose Internet Mail Extensions (MIME). Whereas RFC 822 is limited to handling text messages of a single body part, MIME extends RFC 822 to handle multiple body parts.

Therefore, the content of a message can include text as well as images, audio, video, and binary or application-specific files. The text included in the message content has the following characteristics:

- Unstructured or structured
- Unlimited line length or overall length
- Non-ASCII character sets, which allows non-English language text
- Multiple fonts

If included, the images, audio, video, and binary or application-specific files appear as attachments.

MIME defines the following headers that can appear at the start of an Internet mail message:

- **MIME-version**—Specifies a version number to indicate that a message format conforms to the MIME standard.
- **Content-type**—Specifies the type or subtype of data in the content or body of a message. Possible values include the following:
  - **Text**—Indicates data that is principally text.
  - **Multi-part**—Indicates a message consisting of multiple body parts, each having its own data type.
  - **Application**—Indicates either application or binary data.
  - **Message**—Indicates an encapsulated message.
  - **Image**—indicates still image (picture) data.
  - **Audio**—Indicates audio or voice data.
  - **Video**—Indicates video or moving image data, possibly with audio as part of the composite video data format.
- **Content-transfer-encoding**—Specifies how the data is encoded so that the data can traverse Internet Message Transfer Agents (IMTAs) outside of the SIMS email system that may have data or character set limitations.
- **Content-ID**—Specifies an ID for a message content or body.
- **Content-description**—Text that provides descriptive information about the message content or body.

For complete information on MIME, refer to RFCs 1521, 2045, 2046, 2047, 2048, and 2049.

IMTA Directory Cache

The delivery and routing of messages by the Internet Message Transfer Agent (IMTA) is based on the user and group (distribution list) entries stored in the directory service. The IMTA needs to access the directory information for each message that it processes. Rather than querying the directory service each time it processes a message, the IMTA *caches* the directory information, or takes a snapshot of the directory information and stores it, and accesses directory information in the cache. The IMTA implements the cache for the following reasons:

- **Performance**—Performing a directory query for each recipient of each message is time-consuming and puts a large load on the mail server.
- **Data formatting**—The information stored in the directory service is not always in the format needed by the IMTA. When creating the cache, the IMTA reformats the directory information.

The directory information stored in the directory service is continuously updated. As a result, the directory information in the IMTA-directory cache must be updated periodically with the current directory information in the directory service or *synchronized*. Two types of synchronization are supported:

- **Full synchronization**—The existing cache is replaced with a new cache, completely rebuilt with the current user and group entries from the directory service. After the synchronization occurs, the IMTA configuration file is rebuilt then the IMTA is automatically restarted.
- **Incremental synchronization**—The existing cache is updated with user and group entries that were created or modified since the last full or incremental synchronization. The IMTA is not restarted.

TABLE 7-1 outlines the updates to the IMTA-directory cache that are and are not performed during a full synchronization and an incremental synchronization.

TABLE 7-1 Updates Performed During Full/Incremental Synchronizations

IMTA-Directory Cache Update	Performed During Full Synchronization?	Performed During Incremental Synchronization?
New user entries added	Yes	Yes
Modified user entries updated	Yes	Yes
Deleted user entries removed	Yes	No
New members added to existing distribution lists	Yes	Yes
Deleted members removed from existing distribution lists	Yes	Yes
Modification of access control info	Yes	No
New distribution lists added	Yes	Yes
Deleted distribution lists removed	Yes	Yes

Address Rewrite Rules

When a message enters an Internet Message Transfer Agent (IMTA) channel, it must be placed in the correct channel queue and subsequently routed to the correct destination for delivery. The *domain rewriting rules* (from this point forward called the *rewrite rules*) are a set of tool that the IMTA uses to route messages to the correct host. Rewrite rules perform the following functions:

1. **Extract the host/domain specification from an address of an incoming message**
2. **Match the host/domain specification with a rewrite rule pattern**
3. **Rewrite the host/domain specification based on the domain template**
4. **Decide the IMTA channel queue in which the message should be placed**

The following sections walk you through the rewrite rule process. They also discuss the following elements of the rewrite rule itself:

- **Pattern**—A string composed of ASCII characters that the host/domain specification can potentially match
- **Domain templat**e—A template that defines how the host/domain specification is rewritten
- **Routing system**—The destination channel
- **Controls**—Sequences that impose conditions to the applicability of a rewrite rule.

The address may be returned as one of the following:

- If the address returned is `jdoe@sims-ms.myhost.bridge.net`, the domain part matches a rule whose routing system is that of the message store channel. The message is then queued to the `sims-ms` channel and delivered to the store by the channel's master program.
- If the address returned is `jdoe@myhost.bridge.net`, the domain part matches a rule in the `l` (local) channel section. If the address matches the address previously looked up in the alias table, the destination channel is set to `l` (`/var/mail`) and the message is enqueued. If it does not match, the same operation is reiterated until it stabilizes on one address, the limit of the alias lookup is reached, or a self reference (a:b, b:c, c:a) is found. In the last two cases, the address is not resolved, which causes the messages to be bounced (unknown user error message). The alias lookup default is 10. This value can be changed by configuring the `MAX_ALIAS_LEVELS` option in the option file.
- If the address returned is `jdoe@host2.bridge.net`, the domain part matches a rule in the Intranet SMTP channel section and the message is enqueued to the `tcp_local` channel (intranet channel), and so on.
- If the string `jdoe` does not match anything in the alias cache, the message is returned to the originator with the "user unknown" error.

Extracting the Host/Domain Specification of An Address

When a message enters a channel, all addresses on the envelope and in the message header are examined and the host/domain specification of the address is extracted. The host/domain specification is the part of the address that is to the right of the at (@) sign. For example, in the address `john@stream.bridge.net`, `stream.bridge.net` is the host/domain specification.

Matching Host/Domain Specification with a Rewrite Rule Pattern

The channel scans for a match between the extracted host/domain specification and the pattern portion of the first rewrite rule in the list. (The channel scans the host/domain specification from left to right—for example, starting with `corp`, then `bridge`, then `net`—and the rewrite rules list from top to bottom.) If a match is not found in the first rule, the channel scans the next rule and so on. If a match is found, the host/domain specification is rewritten per the domain template of the rewrite rule. If a match is not found, the message is returned to the sender.

TABLE 7-2 outlines the types of rewrite rule patterns that your rewrite rule list can potentially contain and the order in which each type of pattern is scanned when the input address is `joe@sc.cs.bridge.net`. The rewrite rule pattern types are listed in the order in which the pattern is scanned.

TABLE 7-2 Rewrite Rules Pattern Types

Example of Pattern Scanned For	Explanation
sc.cs.bridge.net	matches sc.cs.bridge.net only
*.cs.bridge.net	matches <any one token>.cs.bridge.net only
.cs.bridge.net	matches <multiple tokens>.cs.bridge.net only
..bridge.net	matches <any one token>.<any one token>.bridge.net only
.bridge.net	matches <multiple tokens>.bridge.net only
..*.edu	matches <any one token>.<any one token>.<any one token>.edu only
.edu	matches <multiple tokens>.edu only
..*.*	matches <any one token>.<any one token>.<any one token>.<any one token> only
.	matches anything

Note – Your channel does not necessarily contain a rewrite rule and, subsequently, a rewrite rule pattern for each of the patterns described in TABLE 7-2. However, if your channel does contain a rewrite rule for each of the described patterns, then TABLE 7-2 outlines the default order in which each pattern is scanned for until a match is found. That is, in whatever the order the rules are written, they are scanned from most specific to least specific.

The rewrite rule list for a channel can include multiple rewrite rules containing the same pattern but different domain templates and control sequences. You can reconfigure the order of these types of rewrite rules so that the channel scans these rules in the reconfigured order. This features enables you to fine-tune the rewrite rule list with your preference as to how the domain/host specification in these types of rules is rewritten. You cannot reconfigure the basic order in which the rewrite rules in TABLE 7-2 are scanned.

To illustrate how the host/domain specification and rewrite rule matching process works, if the extracted host/domain specification is `zoo.bridge.net` and the `eng.bridge.net` and `*.bridge.net` patterns are defined in the rewrite rules, then a match would result after the second scan with *.bridge.net.

Rewriting the Host/Domain Specification

If a match is made between the host/domain specification of an address and the pattern portion of a rewrite rule, the host/domain specification is rewritten according to the domain template portion of the rewrite rule.

The *domain template* defines how the host/domain specification is rewritten. The template defines how to rewrite the domain port. It does that either statically or dynamically depending on the domain.

Mapping a Rewritten Address to a Destination Channel

The last element of a host/domain rewrite rule is the *routing system* or destination channel in whose queue a message should be placed for delivery.

Rewrite Rule Controls

By default, a rewrite rule is scanned for all header and envelope addresses. It is possible to specify that a given rule applies to a subset of all the addresses. For example, a rule can search for some combination (AND) of the *To:*, *From:*, *CC:*, and *Bcc:* header addresses and the envelope recipient addresses (RCPT TO). Additional filters (control sequences) are available and are documented in the *un Internet Mail Server 4.0 Reference Manual*.

Default Channel Rewrite Rules

During installation of SIMS, the system generates default rewrite rules for each of the installed channels. You can modify some of the default rewrite rules or add new rewrite rules for each of the installed channels using the Rewrite Rules section in the IMTA property book of the Administration Console.

Adding and Reconfiguring Rewrite Rules

The Administration Console enables you to add a rewrite rule to an existing or newly created channel. It also enables you to delete or modify an existing rewrite rule associated with an existing channel.

SMTP Authentication

The SIMS SMTP authentication feature allows only password authenticated users to perform SMTP relaying. That is, only users with accounts and passwords in the SIMS directory can use the SMTP relaying mechanism. This prevents anonymous spammers from using the SMTP relay to send UBE.

RFC 2554 defines an extension of the SMTP protocol which provides the ability to authenticate clients and server. It uses the SASL (Simple Authentication and Security Layer) framework defined in RFC 2222. This functionality is commonly known as SMTP AUTH, owing to the `AUTH` command added to SMTP as part of this extension.

Background and Terminology

An *authentication mechanism* is a particular method for a client to prove its identity to a server. APOP, PLAIN and KERBEROS4 are examples of authentication mechanisms. SIMS initially supports only the PLAIN SASL mechanism. More mechanism are like likely to become supported ones in a near future. The PLAIN SALS mechanism is defined in
`http://search.ietf.org/internet-drafts/draft-newman-tls-imappop-09.txt`

An *authentication verifier* (for example, a password) is stored on the server and contains information used to verify a user's identity. The format of the authentication verifier may restrict which mechanisms can be used. The term authentication verifier is preferred in place of password, because while passwords are the most common instance of authentication verifiers, an authentication verifier could also be something like a certificate in an LDAP directory; usually, however, you may think *password* wherever you see *authentication verifier.*

An *authentication source* is a file, database, or interface to an LDAP directory, where the authentication verifiers for users are stored. The authentication source must be accessible to the server. SIMS uses the user passwords stored in the LDAP directory as its authentication source.

SASL is a framework for adding different authentication mechanisms to Internet protocols such as POP, IMAP, and SMTP. When the connection is opened, the POP, IMAP, or SMTP client may authenticate itself to the respective server.

Controlling SMTP Email Access

An individual can intentionally or inadvertently overwhelm your mail server by flooding it with messages. This type of act is called a *denial of service attack.* If a denial of service attack is perpetrated against your mail server, either there may be a substantial impact to the throughput of your mail server or your mail server will become overloaded and nonfunctional.

SIMS provides features that enable you to minimize the possibility of a denial of service attack as well as help you control spam (unwanted or unsolicited email):

- **Email access restrictions**—Enables you to specify which incoming messages are accepted or denied based on recipient or originating domain, client IP address, server IP address, or originating email address.
- **Message size limits**—Enables you impose a limit at which a message is deemed too large and rejected by a channel.

Distribution Lists

The directory service stores and manages distribution lists as group entries. Each group entry is composed of attributes, for example, distribution list name, members, and so on. For example, imagine that the widget team of the Bridge Corporation is:

- Jane: `jane@eng.bridge.net`
- Bernie: `bernie@eng.bridge.net`
- Kevin: `kevin@eng.bridge.net`
- Amy: `amy@eng.bridge.net`
- Frank: `frank@eng.bridge.net`

You can configure a distribution list composed of each widget team member’s email address along with the associated email address `widget@eng.bridge.net`. In addition to distribution list members, you must also configure a distribution list owner. An *owner* is an individual who is responsible for the distribution list. An owner can add or delete distribution list members.

You can optionally configure a distribution list moderator. A *moderator* is an individual, usually the distribution list owner, who initially receives a message addressed to a distribution list. Upon receipt of a message, the moderator can forward the message to the distribution list, edit the message and then forward it, or not forward the message.

You can specify distribution list Access Control; that is, what domains and users can or cannot distribute mail to the group. By default, if a moderator is created, only the moderator can send mail to the group and all other submissions go to the moderator. If no moderator is specified, then anyone can send mail to all the group members.

The *Send Error Conditions To* field enables you to specify an individual to receive a message if an error condition with a distribution list arises. An example of an error condition is when a message addressed to the distribution list cannot be delivered. Upon receipt of one of these messages, it is the specified individual’s responsibility to notify the email administrator about the error. (You can specify the email administrator as the recipient of these messages.)

The *Send Request Messages To* field enables you to specify an individual to receive messages containing requests to be added as a distribution list member. Upon receipt of one of these messages, it is the specified individual’s responsibility to notify the email administrator about adding the requestor as a distribution list member. Requests are sent to `<list name>-request@domain`. (As with the *Send Error Conditions To* field, you can specify the email administrator as the recipient of these messages.)

If you don’t configure the *Send Error Conditions To* and *Send Request Messages To* fields, then by default, the individual who is configured as the owner of the distribution list will receive the respective messages.

Distribution List Process

The Internet Message Transfer Agent (IMTA) retrieves the group entry attributes from the directory service (via the IMTA-directory cache) when it processes a distribution list. The distribution list process is composed of the following phases:

- Routing if necessary
- Address processing
- Message expansion
- Routing or delivery

Imagine that the Bridge Corporation employee Steve, who is in the finance department, sends a message to distribution list `widget@eng.bridge.net`. The widget distribution list is composed of a group of fellow Bridge Corporation employees, who develop widgets in the engineering department.

The first phase in the distribution list process applies only to email systems that contain multiple mail servers and therefore IMTAs. If your email system has only one IMTA, then this phase does not apply. The IMTA that accepts Steve's message determines if the distribution list `widget@eng.bridge.net` is stored as a group entry on this particular mail server. If not, the IMTA checks its data base to determine which IMTA in the email system stores this particular group entry. If the IMTA finds this information in its data base, the message is routed to the appropriate IMTA. If the IMTA does not find the information in its data base, the message is routed to the IMTA configured as the router or smart host in that particular domain. Once the message is accepted by the mail server that stores the group entry for widget@eng.bridge.net, the IMTA processes each distribution list member's address (rewrite rules, access control, and so on).

The IMTA then *expands* the message or converts the message into enough copies for each distribution list member. Rather than converting the message into one copy per distribution list member, the IMTA converts the message into one copy per IMTA channel that will deliver or route the message to a distribution list member. For example, if two distribution list members have mailboxes in `/var/mail` and two in the Sun message store, the IMTA will convert the message into two copies: one for the `/var/mail` channel and one for the Sun message store channel.

The IMTA then places a copy of the message in the queue of each IMTA channel that will deliver or route the message to a distribution list member.

Distribution Lists can be configured to accept or reject messages from certain users or domains. The detailed access control for this is explained in the *Sun Internet Mail Server 4.0 Reference Manual.*

Access Control

The Sun Internet Mail Server supports a distribution list access control feature. If you do not implement the access control feature, any email user who knows the email address can send messages to a particular distribution list.

The access control feature enables you to configure who can send messages to a particular distribution list, thereby controlling the quantity of messages as well as the quality of information that can be sent to a distribution list. You can control access to a distribution list in the following ways:

- **Submitter**—You can specify a list of submitters (users or groups) who are authorized to send messages to a particular distribution list and/or a list of submitters (users or groups) who are unauthorized.
- **Domain**—Specify a list of domains authorized to send messages to a particular distribution list and/or a list of domains that are unauthorized to send messages to a particular distribution list.

You can specify submitters who are within the email system that you are configuring or in a different email system. The four possible configured access control lists are examined in the following order:

1. Unauthorized submitter list
2. Authorized submitter list
3. Unauthorized domain list
4. Authorized domain list

FIGURE 7-1 shows the access control process.

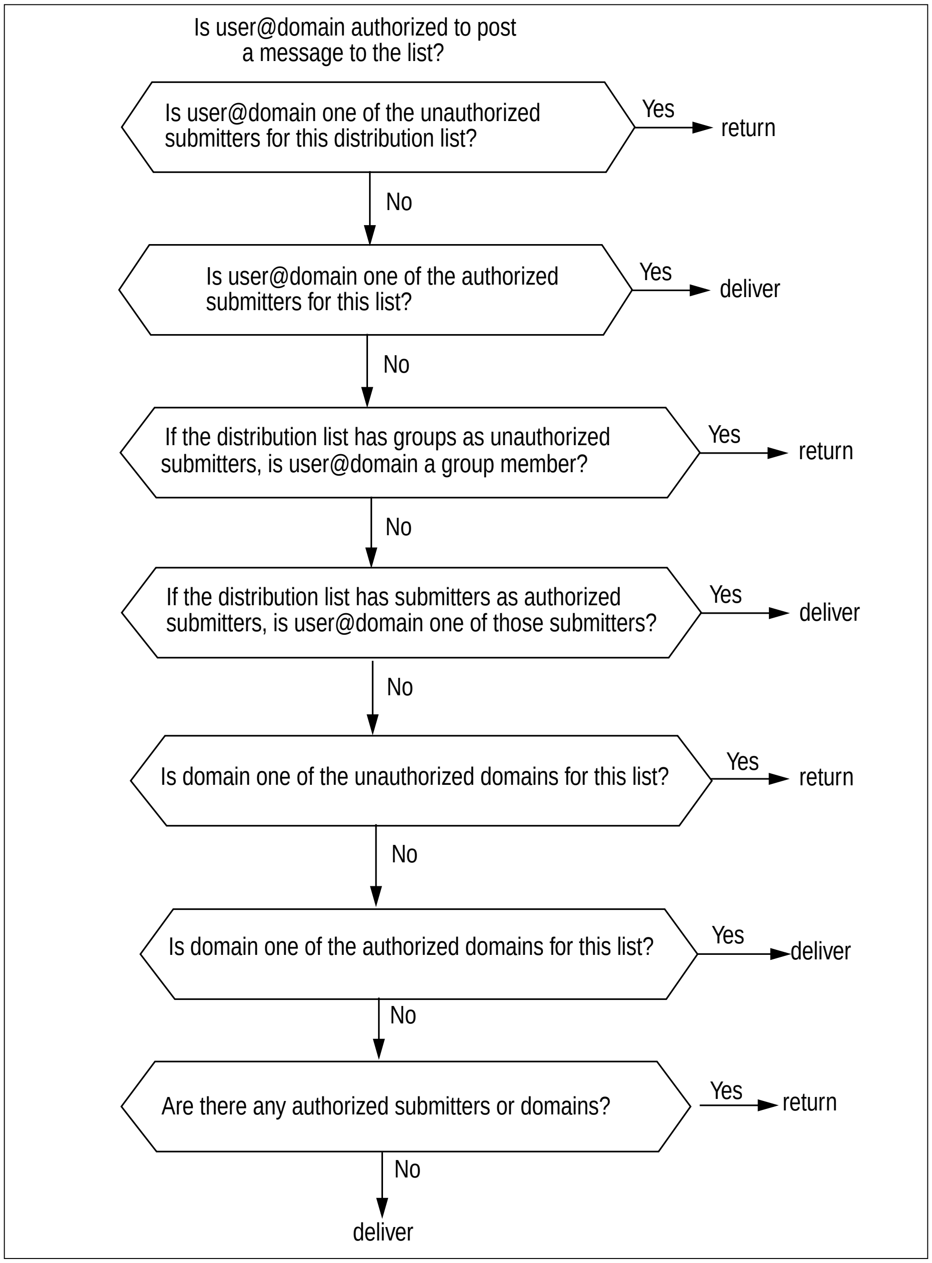

Is user@domain authorized to post
a message to the list?
Is user@domain one of the unauthorized submitters for this distribution list?
Yes
return
No
Is user@domain one of the authorized submitters for this list?
Yes
deliver
No
If the distribution list has groups as unauthorized submitters, is user@domain a group member?
Yes
return
No
If the distribution list has submitters as authorized submitters, is user@domain one of those submitters?
Yes
deliver
No
Is domain one of the unauthorized domains for this list?
Yes
return
No
Is domain one of the authorized domains for this list?
Yes
deliver
No
Are there any authorized submitters or domains?
Yes
return
No
deliver

FIGURE 7-1 Distribution List Access Control Process

If a conflict arises between your configured unauthorized or authorized access control lists, the restriction configured in the submitter list will take precedence over the restriction configured in the domain list. For example, imagine that submitter steve@finance.bridge.net attempts to send a message to the distribution list widget@eng.bridge.net. If the domain finance.bridge.net is on the unauthorized domain list but user steve@finance.bridge.net is on the authorized submitter list, then the message will be delivered because of the precedence of the submitter lists over the domain lists.

TABLE 7-3 contains various ways that you may want to use the distribution list access control feature and the suggested way to achieve the results you want.

TABLE 7-3 Implementing Distribution List Access Control Feature

Restrictions/Allowances You Want to Impose	Suggested Method of Implementing*
Restrict all submitters from a particular domain from sending messages	Enter unwanted domain on unauthorized domain list.
Allow all submitters from a particular domain to send messages	Enter domain on authorized domain list.
Restrict one or more submitters from a particular domain from sending messages and allow all other submitters in the domain to send messages	Enter unwanted submitters on unauthorized submitter list. Enter the domain itself as authorized domain.
Allow one or more submitters from a particular domain to send messages and restrict all others from sending messages	Enter submitters in the authorized submitter list. Enter the domain itself in the unauthorized domain list.
Restrict one or more submitters regardless of domain from sending messages	Enter unwanted submitters in the unauthorized submitter list.
Allow one or more submitters regardless of domain to send messages	Enter submitters in the authorized submitter list.
Restrict submitters to distribution list members only.	Enter distribution list name in the authorized submitter list.

*Although each restriction / allowance outlined in this table can be handled in multiple ways, the documented suggested methods are the most efficient and require the least impact on performance. For example, you can restrict all submitters from a particular domain from sending messages by entering the unwanted domain on the unauthorized domain list or by not including the unwanted domain on the authorized list and having other authorized entries. However, if you restrict this domain by the latter method, the mail server has to go through two steps in the distribution list process rather than one. Therefore, the latter method is not suggested.

CHAPTER 8

Directory Services

Sun™ Internet Mail Server™ 4.0 supports both the Netscape Directory Services (NSDS) 4.1 as well as Sun Directory Services (SunDS) 3.1. NSDS, however, is the preferred directory server that you could use with SIMS 4.0 in the SPARC/Solaris operating environment.

The directory service that SIMS support is based on the Lightweight Directory Access Protocol (LDAP).

Topics in this chapter include:

- What is the Directory Service?
- The Directory Information Tree (DIT)
- The LDAP directory
- Password management
- Sun directory services replication

What is the Directory Service?

The Directory service is a central repository for meta-information, where user profiles, distribution lists, and other system resources information reside.

Although SIMS 4.0 uses LDAP Version 3 (v3)., LDAP v2 and v3 may be used interchangeably during upgrade from SIMS 3.x to SIMS 4.0.

LDAP referrals are a supported feature of LDAP v3. They refer to an LDAP entry that contains a symbolic link (referral) to another LDAP entry. An LDAP referral consists of a machine and a distinguished name. LDAP referrals are often used to reference existing LDAP data so that this data does not have to be replicated. It is also used to maintain compatibility for programs that depend on a particular entry that may have been moved.

To utilize the LDAP referrals, SIMS must first bind using LDAP v3 and then configure the LDAP API to follow referrals transparently if they are encountered during an LDAP operation.

Another LDAP feature supported in SIMS is LDAP server failover, which is basically a backup mechanism for LDAP servers; that is, if one LDAP server fails, the system can switch over to another LDAP server.

SIMS 4.0 supports the following directory services architecture:

- **Distributed architecture**—This makes the directory particularly robust. Even if one server goes down, the other servers can provide access to the directory information.
- **Logical unique database structure**—This allows users and applications to access the same directory information from anywhere on the network. This feature allows decreasing the system administration load. The information only needs to be added, edited, or deleted once, at one point in the network. Millions of entries can be stored.

How Do SIMS Components Use the Directory Services?

SIMS uses the Directory Services to store information commonly used by SIMS Internet Message Transfer (IMTA) and message store and Message Access (MSMA) components.

The IMTA uses the directory to store and retrieve user and distribution routing information as well as user and distribution list profiles.

The MSMA uses the directory to verify the credentials of users and to store user message store configuration information.

The Directory Information Tree (DIT)

In SIMS 3.5, the Directory Information Tree (DIT) supported two types of trees: the Organization tree (OSI tree) and the Domain Component (DC) tree. If SIMS 3.5 were installed at the SP `bridge.net`, the root of these trees was mapped as `o=bridge,c=us` for the OSI tree, and as `dc=bridge,dc=net,` and `o=internet` for the DC tree.

Similarly, if `bridge.net` was hosting the `stream.com` domain, the OSI tree would map this domain as an additional root as `o=stream`, `c=us`. FIGURE 8-1 shows the OSI and DC DIT in SIMS 3.5 for this example.

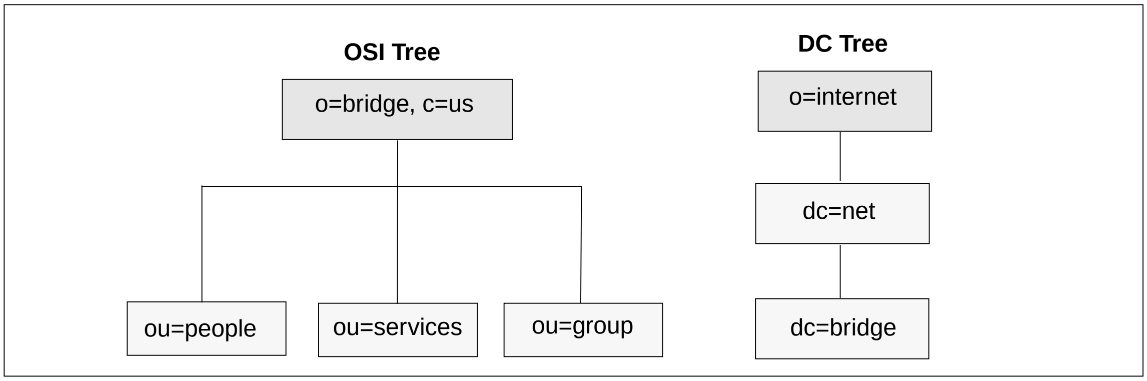


FIGURE 8-1 OSI and DC Tree Representations in SIMS 3.5

In SIMS 4.0, however, the `stream.com` hosted domain must be created in the DIT by nodes `dc=stream,dc=com,` and `o=internet`. These nodes are new roots. FIGURE 8-2 shows the new SIMS DIT structure for the `bridge.net` SP and the `stream.com` hosted domain.

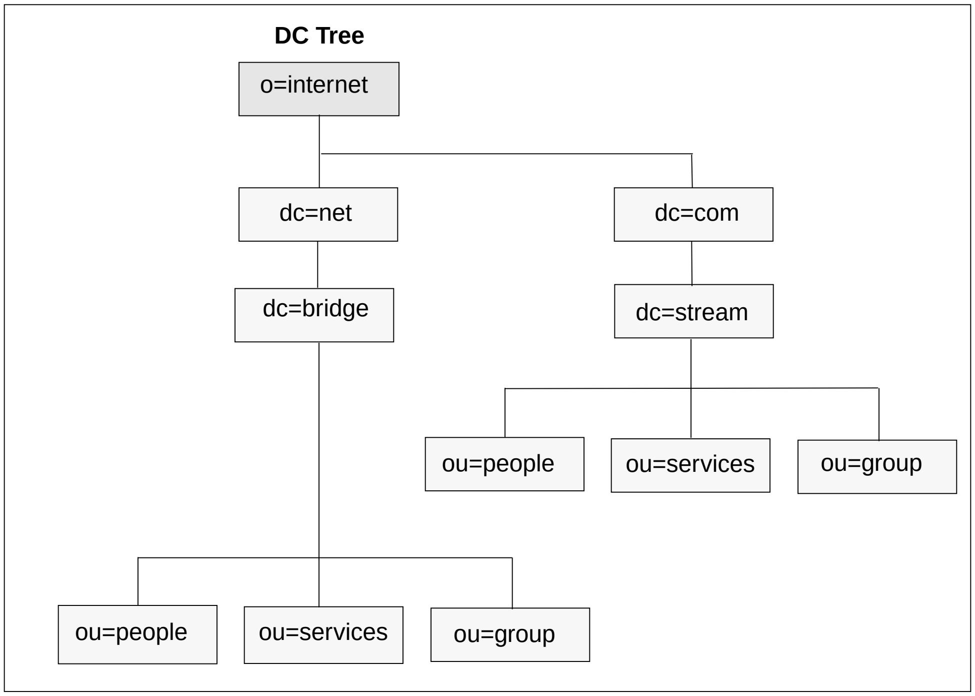


FIGURE 8-2 The DC Tree Representation in SIMS 4.0

Directory Entries

Directory information is stored as directory entries. A *directory entry* is a set of *attributes* and their *values*. For example, SIMS specifies an attribute called `commonname`, its value would be the user's full name.

Directory entries are defined by an *object class* which specifies the kind of object the entry describes and the set of attributes it contains. For example, SIMS specifies an `inetMailUser` object class which has attributes such as `commonname`, `mailHost`, and `mailQuota`. SIMS also specifies an `inetMailGroup` object class which has attributes such as `commonname`, `authorizedSubmitter`, and `rfc822mail,` and `UniqueMemeber`.

The set of object classes supported by a directory service is called the *directory schema*. The schema specifies all the objects and attributes supported by the directory service, as well as which attributes are mandatory and optional for a given object class.

Directory entries are organized hierarchically in a tree-like structure called the *Directory Information Tree* or *DIT*. Each entry has a parent entry and can have child entries. The top of the hierarchy is known as the *root entry*.

An entry is identified by its *distinguished name* (DN). A distinguished name is a sequence of locale-related attributes and values. The first attribute and its value provide the entry's *relative distinguished name* (RDN). The rest of the sequence is the distinguished name of the parent entry. A distinguished name is unique throughout the whole directory service.

FIGURE 8-3 shows an example of how directory information is structured, with the DNs and RDNs of the shaded entries.

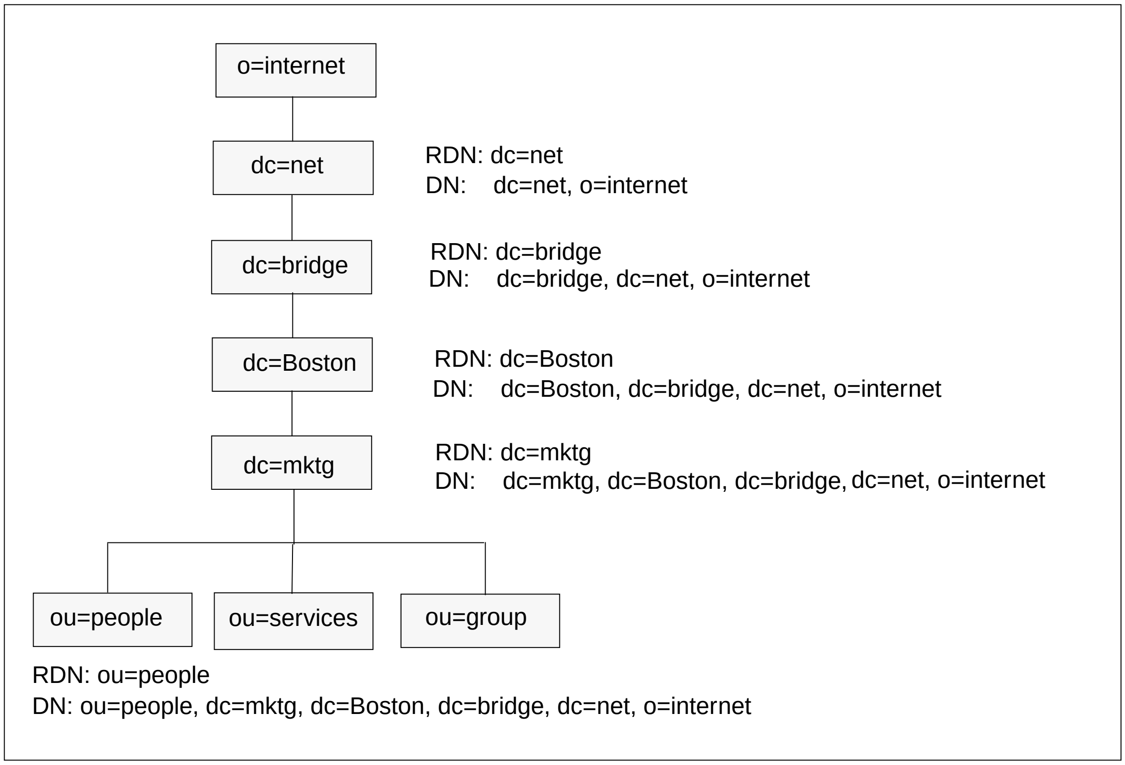

o=internet
dc=net
RDN: dc=net
DN: dc=net, o=internet
dc=bridge
RDN: dc=bridge
DN: dc=bridge, dc=net, o=internet
dc=Boston
RDN: dc=Boston
DN: dc=Boston, dc=bridge, dc=net, o=internet
dc=mktg
RDN: dc=mktg
DN: dc=mktg, dc=Boston, dc=bridge, dc=net, o=internet
ou=people
ou=services
ou=group
RDN: ou=people
DN: ou=people, dc=mktg, dc=Boston, dc=bridge, dc=net, o=internet

FIGURE 8-3 SIMS Directory Information Tree (DIT) Example

User Entries

A user entry or *user profile* contains information on a user. TABLE 8-1 describes each user entry field and whether you must configure a particular field when creating a user entry. A required field is one that actually configures some aspect of the SIMS, while a field that is not required is one that provides incidental information or configures an optional feature.

TABLE 8-1 User Entry Fields

Field	Required/Not Required For SIMS Configuration	Description
Personal Information/Name		
Distinguished name (dn)	Required	A unique path name associated with a user entry that reflects the hierarchy of the directory information tree.
Full name	Required	Stores the possible variations of the first name, last name, and middle initial fields combined. The middle initial is optional. Examples of full names for one particular user are Harrison Green, Harry Green, and Harry A. Green.
First Name	Not required	For example, in the case of Harry Green, the first name is Harry.
Last Name	Required	A last name is a surname, for example, in the case of Harry Green, the last name is Green.
Middle Initial	Not Required	The middle initial is the first letter of the middle name, for example, in the case of Harry A. Green, the initial is A.
Title	Not required	A business or personal title, for example, Accountant or Avid Science Fiction Fan, respectively.
Personal Information/Telephone		
Telephone Number	Not required	Can also include extension number.
Fax Number	Not required	Self explanatory.
Pager Number	Not required	Self explanatory.
Mobile Phone Number	Not required	Self explanatory.
Personal Information/Address		
Postal address	Not required	Self explanatory.
Location	Not required	Self explanatory.
Office Number	Not required	Self explanatory.
Personal Information/Miscellaneous		

TABLE 8-1 User Entry Fields *(Continued)*

Field	Required/Not Required For SIMS Configuration	Description
Home Page	Not required	The Uniform Resource Locator (URL) for a home page.
Description	Not required	Self explanatory.
Additional Information	Not required	Self explanatory.
System Information		
Login name	Required	Unique identification (ID) for user, for example, harryg.
Password	Required	Password associated with login name field; can be stored clear (unscrambled) or crypted (scrambled)
Mail Information		
Mail Host	Required	Name of the user's mail server.
RFC822MailAlias: *<recipient>*	Req. if specified Delivery Channel Type is Connectivity channels	Email address that a recipient inside the email system or the local area network (LAN) proprietary system will see when a message from this user is received.
RFC822MailAlias: *<originator>*	Required if specified Delivery Channel Type is one of Connectivity channels	Email address that a recipient outside the email system or the LAN proprietary system will see when a message from this user is received.
Internet Mail	Required if specified Delivery Channel Type is one of Connectivity channels	All valid Internet, Lotus cc:Mail, Microsoft Mail, or IBM PROFS Mail aliases for a user. The IMTA will accept a message addressed to any one of the specified addresses.
Internet Mail Delivery Options	Required if you specified Internet as Delivery Channel Type	Location of user's Inbox. Can be either `/var/mail` or the Sun Message Store. If `/var/mail`, then must specify mailbox directory. Can optionally enable auto reply, program, forward, and append to file features.
Auto Reply Expiration Date	Req. if you enable auto reply feature in Internet Mail Delivery Options	Date that auto reply feature expires.
Auto Reply Mode	Req. if you enable auto reply feature in Internet Mail Delivery Options	Vacation is only mode currently supported.
Auto Reply Subject	Req. if you enable auto reply feature in Internet Mail Delivery Options	Subject line of auto reply message.
Auto Reply Text	Req. if you enable auto reply feature in Internet Mail Delivery Options	Body of auto reply message.

TABLE 8-1 User Entry Fields *(Continued)*

Field	Required/Not Required For SIMS Configuration	Description
AutoReplyText For Use Within Organization	Req. if you enable auto reply feature in Internet Mail Delivery Options	Body of auto reply message for use within the user's organization.
Program Delivery Info	Req. if program feature is enabled in Internet Mail Delivery Options	Specifies one or more commands with arguments to deliver to a UNIX program.
Forwarding Address	Req. if forward feature is enabled in Internet Mail Delivery Options	Internet address to which email should be forwarded.
Delivery File	Req. if append to file feature is enabled in Internet Mail Delivery Options	Pathname of file to which email should be attached to the end of.
Calendar Information		
CalendarHostName	Req. for Web Access calendars	Calendar server host name
Default Calendar	Req. for Web Access calendars	Name of default Calendar.

Group Entries

A group entry contains information for a distribution list. TABLE 8-2 describes each group entry field and whether you must configure a particular field when creating a group entry. A required field is one that actually configures some aspect of the SIMS, while a field that is not required provides incidental information or configures an optional feature.

TABLE 8-2 Group Entry Fields

Field	Req/Not Req. For SIMS Configuration	Description
General info./General		
Distinguished name (dn)	Req.	A unique pathname associated with a group entry that reflects the hierarchy of the directory information tree (DIT).
Full name	Req.	The possible variations of the group address. An example of a full name for one particular group is marketing.

TABLE 8-2 Group Entry Fields *(Continued)*

Field	Req/Not Req. For SIMS Configuration	Description
Mail domain	Req.	The mail domain in which a group's mail server resides, for example, sales.alpha.com.
Send Error Conditions To	Req.	The individual who receives a notice when an error condition related to the distribution list arises, for example, if a message addressed to the distribution list cannot be delivered.
Send Request Messages To	Req.	The individual who receives a notice when another individual requests being added as a distribution list member.
Mail Host	Req.	The hostname of the group's mail server.
Password	Req.	The password associated with a group and with a shared mailbox. It can be stored clear (unscrambled) or crypted (scrambled). You are prompted for this password when attempting to modify group entry attributes using the command line interface or the user administration interface.
General info./Telephone		
Expandable	Not req.	Make list of members for a particular group or distribution list accessible to all users.
Telephone Number	Not req.	Telephone number for the group. Can also include extension number.
Fax Number	Not req.	Fax number for the group.
Pager Number	Not req.	Pager number for the group.
Mobile Phone Number	Not req.	Mobile phone number for the group.
General info./Address		
Postal address	Not req.	Postal address for the group.
Location	Not req.	Location for the group.
Building	Not req.	Building of the group.
Office Number	Not req.	Office number for the group.
Home Page	Not req.	The Uniform Resource Locator (URL) for a home page.
Description	Not req.	Description for the group.
Additional Information	Not req.	Additional information for the group.
Owner		
Owner	Req.	An owner is an individual who is responsible for a distribution list. An owner can add or delete distribution list members.
Moderator		

TABLE 8-2 Group Entry Fields *(Continued)*

Field	Req/Not Req. For SIMS Configuration	Description
Moderator	Not req.	If moderator feature is enabled, a message addressed to a distribution list is initially sent to the moderator only. The moderator can take one of the following actions: forward the message to the distribution list, edit the message and then forward it, or not forward the message.
Member Information		
UniqueMember	Not req.	A member is a user or group who receives a copy of an email addressed to a distribution list.
Additional Delivery Options		
Shared Mailbox	Not req.	Specifies that messages are delivered to a shared mailbox in the Sun Message Store.
Program	Not req.	Specifies one or more commands with arguments to deliver to a UNIX program.
Append to File	Not req.	Path name of file to which email should be appended (attached to the end of).
Access Control		
Authorized Domain	Not req.	Domain name from which users or groups are authorized to send messages to a particular distribution list.
Unauthorized Domain	Not req.	Domain name from which users or groups are not authorized to send messages to a particular distribution list.
Authorized Submitter	Not req.	Name of user or group who are authorized to send messages to a particular distribution list. If the user or group is internal to the email system, specify the distinguished name; if external to the email system, specify an email address in RFC 822 format.
Unauthorized Submitter	Not req.	Name of user or group who are not authorized to send messages to a particular distribution list. If the user or group is internal to the email system, specify the distinguished name; if external to the email system, specify an email address in RFC 822 format.

Password Management

A directory entry for a SIMS user contains a User Password attribute. The value of this attribute, which is used to authenticate the user to the directory, can be stored in encrypted or unencrypted format.

When you supply a password for authentication, or as an attribute value in a directory operation, you specify the value in unencrypted format. You may not enter the password in its encrypted format.

The Netscape Directory Services supports the `sha` encryption method to encrypt passwords, where as the Sun Directory Services 3.1 uses both the `crypt(1)` utility and the `sunds` encryption method to encrypt passwords.

Sun Directory Services Replication

You can share between several directory servers the load of processing requests generated by directory service clients for the same information. This is done by defining a *replica*, or *slave server* to provide an alternative access point to the directory service for clients. A master naming context can have more than one replica naming context. FIGURE 8-4 shows a master server with two replica servers. *Replication* is the process by which changes in the master data store are propagated to all the replica naming contexts. You can replicate an entire naming context, a subtree, or a particular entry. You can replicate the entire content of an entry or you can specify a subset of attributes to be replicated.

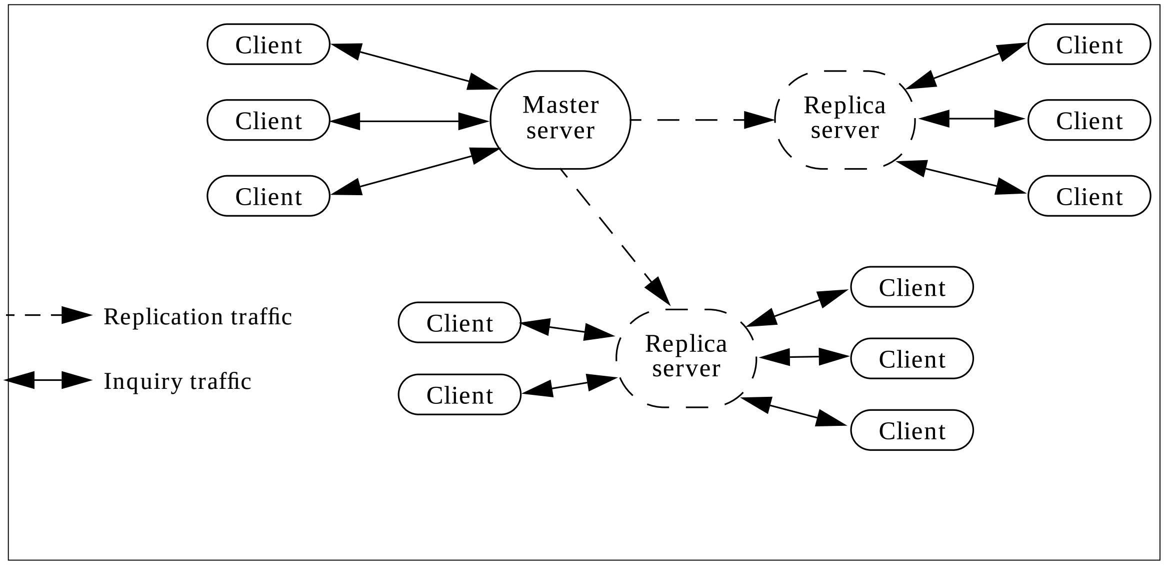


FIGURE 8-4 Master and Replica Servers

Using replication has the following advantages:

- It reduces the load on the master server by diverting some traffic to other servers.
- You can store copies of data where it is mostly frequently used, thus reducing network traffic.
- You can store many copies of the data, but the data is maintained from a central location.
- You need only replicate the data that is required by clients of the replica server, if you know the requirements of those clients specifically enough. You may be able to tailor a replica exactly to the needs of a specific client. By reducing the number of entries replicated, you reduce the network traffic caused by replication updates.
- You could maintain a *public* replica server containing information that is not confidential, allowing greater access to this information than you usually allow for other servers. For example, you could create a server containing the email addresses for the sales and support staff who deal with current products but not the research staff working on future products, and make it available to the sales staff of a partner company.

Note – You could provide the same partial view of directory information with appropriate access controls. However, using a partial replica on a dedicated machine ensures that you are not providing access to your entire network. For extra security, you could connect the replica server to your network only while the replication update is in progress.

The costs of using replication are:

- Additional network traffic caused by replication of data. However, though there may be an overall increase in traffic, more of the traffic will be local, so you can avoid known network bottlenecks for inquiry traffic. Also, you can time the replication updates for when the network is least busy.
- Information retrieved from replicas may be out of date if replication has not happened since an update, so certain applications may always need to query the master data store.
- You cannot modify a replica. All updates must be performed on the master copy of an entry.

How SunDS Replication Works

Information from a master naming context is propagated to a replica by the `slurpd` daemon. The `slurpd` daemon can run permanently, so that updates to directory information are propagated immediately, or you can define a synchronization schedule. You can override the schedule at any time and trigger an immediate synchronization. This is useful if you change a large number of entries and do not want to wait for the next scheduled synchronization.

The `slurpd` daemon uses the LDAP protocol to update a replica naming context. A master naming context for which a replica is defined maintains a replication log. Each time the master naming context is updated, the transaction is recorded in the replication log. When the `slurpd` daemon next runs, it reads the replication log and sends the change to the `slapd` server that holds the replica naming context. The `slapd` server handles update requests from `slurpd` in the same way that it handles all requests, using the information supplied in the bind request to set the access level granted to `slurpd` requests. To guarantee that all replication updates are completed, `slurpd` must bind with the DN defined when the replica naming context was configured. If a different DN is used, write access for all entries may not be granted.

A replica data store always has a referral pointing to the master data store. If a replica server receives a request to modify an entry, it returns a referral to the client, indicating the master server to be contacted. In some cases, the client software handles the referral automatically and the user need not resubmit the query. Once the modification has been made in the master naming context, the change is sent to the replica naming context the next time the `slurpd` daemon runs.

FIGURE 8-5 shows the naming context in the Stream corporation DIT.

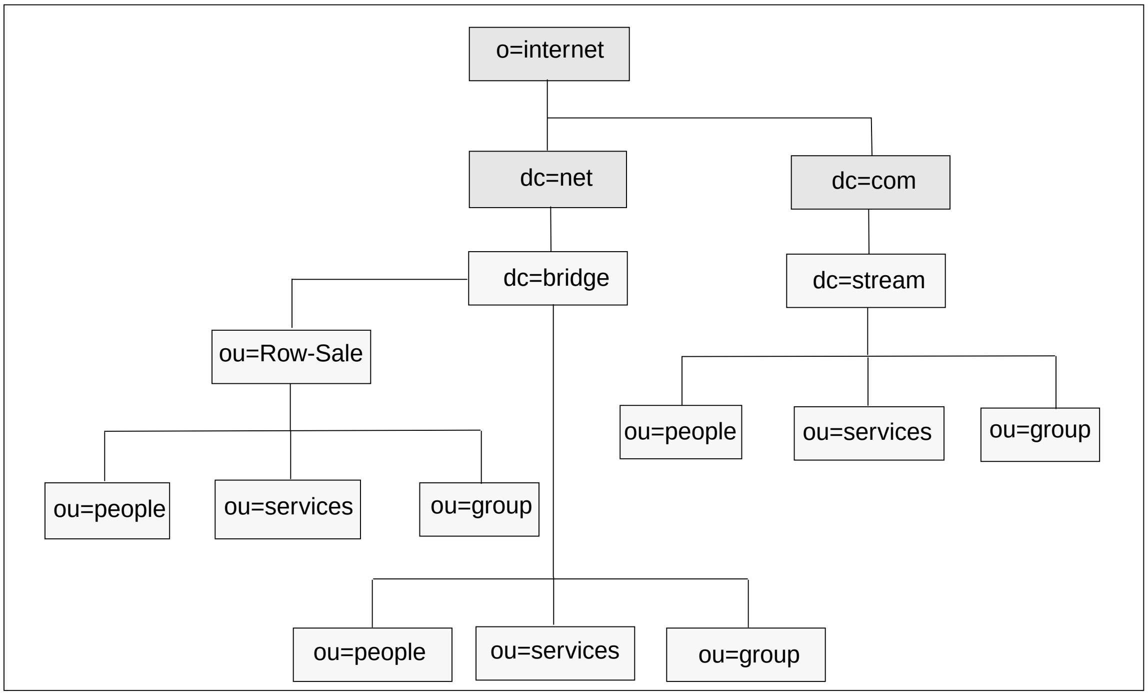


FIGURE 8-5 DIT Naming Context Replica

Example: Replication in the Stream Corporation

The following example characterizes the replica for several naming contexts in the Stream corporation DIT:

- All servers will contain a replica of `l=Boston, dc=bridge, dc=net, o=internet` for fast access to entries concerning the headquarters of the corporation. Only ussales, eursales, and rowsales get their replicas directly from the boston server. The other servers get a replica of the replica from the server that is closest in the network.
- A second server, `eursale2`, will hold a complete replica of `ou=Euro-Sales, dc=bridge, dc=net, o=internet` to share the load on the existing eursales server.
- Each of the servers at the distribution centers will hold complete or partial replicas of the other distribution center naming contexts. For example, the `atlanta` server will hold a complete replica of `ou=London-Dist, dc=bridge,`

`dc=net`, `o=internet` and a partial replica of `l=Tokyo`, `dc=bridge, dc=net`, `o=internet` containing the information about the distribution center but not about the sales office.

TABLE 8-3 shows the replication strategy for each server in the Stream Corporation directory service.

TABLE 8-3 Replication Strategy for the Stream Corporation

Server	Naming Contexts	Replication Status
boston	l=Boston, dc=bridge, dc=net, o=internet	master, replicated to ussales, eursales, and rowsales
ussales	ou=US-Sales, dc=bridge, dc=net, o=internet	master
	l=Boston, dc=bridge, dc=net, o=internet	replica from boston, replicated to atlanta and sanfran
eursales	ou=Euro-Sales, dc=bridge, dc=net, o=internet	master, replicated to eursale2
	l=Boston, dc=bridge, dc=net, o=internet	replica from boston, replicated to eursale2, london, and paris
eursale2	ou=Euro-Sales, dc=bridge, dc=net, o=internet	replica from eursales
	l=Boston, dc=bridge, dc=net, o=internet	replica from eursales
rowsales	ou=Row-Sales, dc=bridge, dc=net, o=internet	master
	l=Boston, dc=bridge, dc=net, o=internet	replica from boston
atlanta	ou=Atlanta-Dist, dc=bridge, dc=net, o=internet	master, replicated to london and tokyo
	l=Boston, dc=bridge, dc=net, o=internet	replica from ussales
	ou=London-Dist, dc=bridge, dc=net, o=internet	replica from london
	ou=dist, l=Tokyo, dc=bridge, dc=net, o=internet	partial replica from tokyo
sanfran	l=San-Francisco, dc=bridge, dc=net, o=internet	master
	l=Boston, dc=bridge, dc=net, o=internet	replica from ussales

Server	Naming Contexts	Replication Status
london	ou=London-Dist, dc=bridge, dc=net, o=internet	master
	l=Boston, dc=bridge, dc=net, o=internet	replica from eursales
	ou=Atlanta-Dist, dc=bridge, dc=net, o=internet	replica from atlanta
	ou=dist, l=Tokyo, dc=bridge, dc=net, o=internet	partial replica from tokyo
lonres	ou=London-RD, dc=bridge, dc=net, o=internet	master
	l=Boston, dc=bridge, dc=net, o=internet	replica from london
paris	ou=Paris-Man, dc=bridge, dc=net, o=internet	master
	l=Boston, dc=bridge, dc=net, o=internet	replica from eursales
tokyo	l=Tokyo, dc=bridge, dc=net, o=internet	master, partially replicated to atlanta and london
	l=Boston, dc=bridge, dc=net, o=internet	replica from rowsales
	ou=Atlanta-Dist, dc=bridge, dc=net, o=internet	replica from Atlanta
	ou=London-Dist, dc=bridge, dc=net, o=internet	replica from london

Sun Directory Services 3.1 Features

SunDS 3.1 offers the following features:

- **Corporate Directories**—A distributed corporate directory, accessible from anywhere in the company, and listing networked resources and users with their related details and attributes is almost essential to the smooth running of an enterprise network. SunDS 3.1 provides the ideal solution for creating such a distributed directory.
- **Global Messaging**—To get the most out of a corporate mail system, it is crucial that users be able to find a correspondent's mail address and profile easily and quickly, from anywhere on the network. SunDS 3.1 enables them to do just that.
- **Naming**—All information on the configuration of the local network, which is used by desktop applications, can be stored in SunDS 3.1.

- **Remote User Authentication**—Network access servers use SunDS 3.1 to authenticate remote users and grant them access to the network.
- **Compatibility** —SunDS 3.1 is a multiprotocol directory server. It supports LDAP v.3, RADIUS, HTTP and NIS. It is also compatible with Windows NT LDAP based applications. SunDS 3.1 implements the RFC 1777 recommendations as an information modeling.

Sun Directory Services 3.1 Components

SunDS 3.1 comprises the following components:

- A multiprotocol server and data store which stores the Directory Information Tree (DIT).
- A replication server which manages data replication between LDAP servers.
- A Java-based configuration and management tool which enables Directory Services to be configured and managed from any Java-compliant Web browser.
- A Web/LDAP gateway enabling users to navigate through the directory, and to query and edit data from and Web browser.
- Database utilities, including a feature which enables import and export of text files from and to other databases.
- An SNMP MADMAN (Management and Directory Management) agent, and a RADIUS SNMP agent.
- A Java-based directory content editor.

CHAPTER 9

Sun Message Store

The Sun Message Store is the primary message store that is used by the Sun™ Internet Mail Server™ 4.0. This provides a significant advance in reliability, performance, and scalability among open systems message stores.

Topics in this chapter include:

- Sun Message Store key features
- How does the message store work?
- User identification for message store and message access
- Mail folders
- Sun OpenWindows Mail Tool V3 format conversion
- Simultaneous connections
- Message access protocols
- Message access service specifications

Sun Message Store's Key Features

The key features of the Sun Message Store component are:

- **Supported Internet Standards**—The message store stores any message that conforms to RFC 822 specifications. It recognizes MIME content format and supports direct address ability of any header or body part. It is specifically designed for IMAP4 message access.
- **Reliable Scalable Design**—Write-once data store and two-level indexing simplify access, reduce contention, and facilitate multithreading. Committed transactions also facilitate multithreading and ensure that no messages are lost or corrupted.
- **High Storage Efficiency**—The message store retains exactly one copy of each message, regardless of the number of recipients.
- **Optimized Access**—Messages are prepared and indexed when inserted into the store. No parsing is necessary when messages are accessed. The degree of preparing is tunable. The benefits of preparing decrease as message size increases. POP users do not need parsing at all.
- **Optimized File System Usage**—Time-based sorting of messages within the data store provides good locality of reference and more effective use of disk caches.
- **Optimized Updates**—Once in the store, messages are never modified. Status changes and folder updates are stored in lightweight index files that are rapidly updated.
- **Managed Backup, Migration, Archival, and Purge**—Bulk dump and load facility supports backup, restore, and migration of individual users, groups, or entire stores. Deletion and purge tools support archival and guaranteed delete.

Component Architecture

The Sun Message Store is organized into the following major subcomponents:

IMTA Delivery Queue

This is the IMTA channel queue interface that accepts messages from the IMTA and inserts them into the message store. The implementation is as follows:

- Messages are appended to the central data store.
- A data store index entry is created.
- A pointer to the index is added to the inbox folder of each recipient.

Retrieval Interface

The retrieval interface is the c-client driver interface, used by c-client applications to manipulate messages in the message store. The Sun Message Store driver can be used only in protocol server applications, like the IMAP4 and POP3 servers.

Backup and Restore Facility

This facility is the generalized bulk dump and load facility that can be used for message store backup, moving users from one message store to another.

Administration

The Sun Message Store includes a store-wide configuration file and user profiles obtained from the Directory.

Monitoring

The message store provides extensive statistics, including disk space in use, number of messages in folders, oldest messages, and most recent activity. Monitoring parameters are exposed through the c-client driver interface. The set of managed objects is specified by the message access component.

Maintenance Utilities

These command-line utilities are provided for periodic maintenance.

How Does the Message Store Work?

The Sun Message Store maintains only one copy of each message. If the Sun Message Store receives a message addressed to multiple users or a group (distribution list), it adds a reference to the message in each user's Inbox rather than having a copy of the message in each user's Inbox, thereby saving disk space. The individual message status (new, unread, replied to, deleted, and the like) is maintained per Mailbox.

FIGURE 9-1 shows the major elements of the Sun Message Store and the flow of messages into the Sun Message Store from the IMTA and to the mail client users.

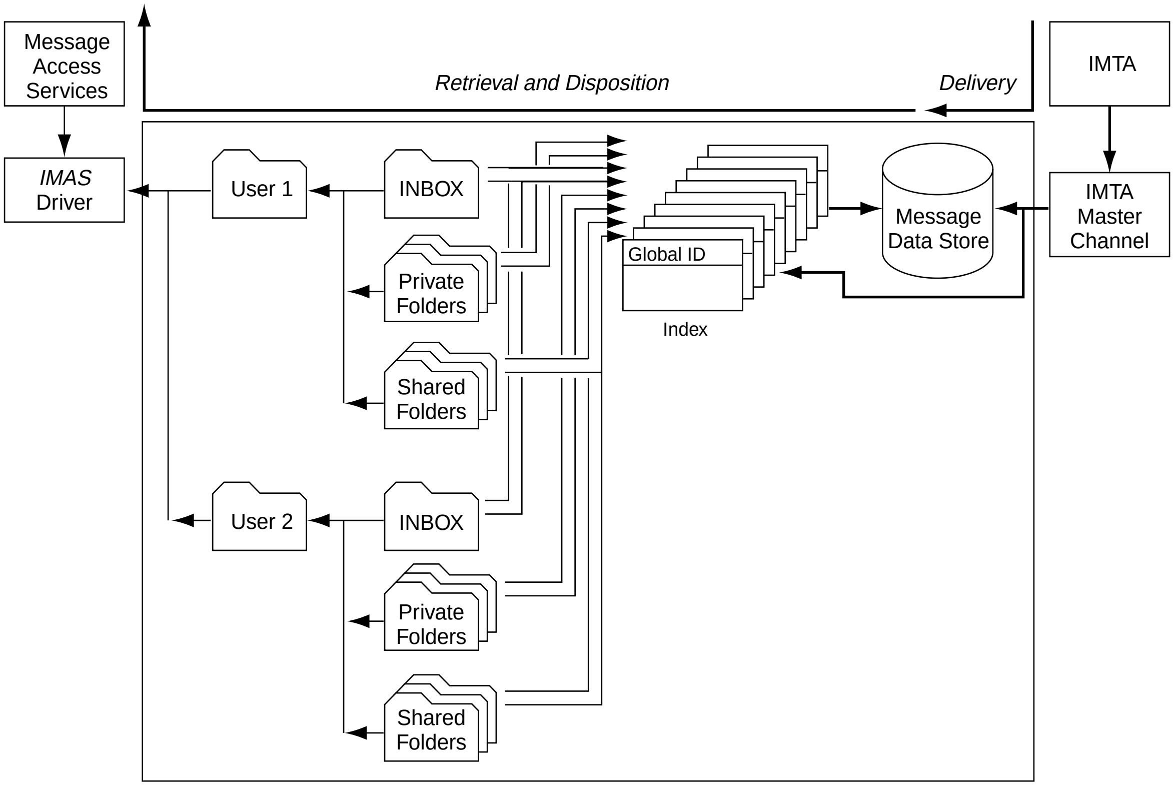


FIGURE 9-1 Sun Message Store Message Flow

SIMS also supports `/var/mail`. A site can implement both the Sun Message Store and `/var/mail`. Mail users can access both the Sun Message Store and `/var/mail` using the Internet Mail Access Protocol version 4 (IMAP4) or the Post Office Protocol version 3 (POP3).

Access to the Sun Message Store and `/var/mail` is multithreaded. This feature enables a single process to manage a large number of connections. Each connection is handled by a thread. Multithreaded access maximizes both performance and scaleability by minimizing the system resources required for the management of each connection.

User Identification for Message Store and Message Access

Each SIMS email user has three distinct identification (ID) strings:

- **Email ID**—The ID used to send mail to this users. It is also called the mail address. Example: `mayflower@bridge.net`
- **Login ID**—The ID a client uses to access his mailbox. It is also called the mail access ID. Examples: `mayflower`, `bluebell+stream.com`
- **Message Store ID**—The user identification required by message store commands such as `imbackup`, `imrestore`, `imcheck`, `iminitquota`, `imexpire`, `imimportmbox`, `imdeluser`, and so on.

The login ID is different from the message store ID depending on whether the LDAP attribute `simsRecursive` is set to 1 or 0 and the user entry is in a subdomain of the default domain. If `simsRecursive` is set to 1, the message store ID of users in a subdomain do not need to be appended with the domain. If `simsRecursive` is set to 0, the Login ID and message store ID of users in a subdomain need to be appended with the domain. In a hosted domain, both the Login ID and message store ID of users in a subdomain need to be appended with the domain. In either case, Email IDs do not change. Below are three examples:

Example 1, default domain: `stream.com`, simsRecursive=1

Distinguished Name for mayflower:
`cn=mayflower,ou=People,dc=stream,dc=com,o=internet`
Distinguished Name for bluebell:
`cn=bluebell,ou=People,dc=eng,dc=stream,dc=com,o=internet`

TABLE 9-1 Default Domain: `stream.com`, simsRecursive=1

Message Store ID	Login ID
mayflower	mayflower or mayflower+stream.com
bluebell	bluebell or bluebell+eng.stream.com

Example 2, default domain: `stream.com`, `simsRecursive=0`:

Distinguished Name for mayflower:
`cn=mayflower,ou=People,dc=stream,dc=com,o=internet`
Distinguished Name for bluebell:
`cn=bluebell,ou=People,dc=eng,dc=stream,dc=com,o=internet`

TABLE 9-2 Default Domain: `stream.com`, simsRecursive=0

Message Store ID	Login ID
`mayflower`	`mayflower, mayflower+stream.com`
`bluebell@eng.stream.com`	`bluebell+eng.stream.com`

Example 3, Hosted domain:

Distinguished Name for mayflower:
`cn=mayflower,ou=People,dc=string,dc=com,o=internet`
Distinguished Name for bluebell:
`cn=bluebell,ou=People,dc=eng,dc=stroller,dc=com,o=internet`

TABLE 9-3 Hosted Domain

Message Store ID	Login ID
`mayflower@string.com`	`mayflower, mayflower+string.com`
`bluebell@stroller.com`	`bluebell+stroller.com`

Mail Folders

The Sun Message Store is where user messages are stored and retrieved. Each user has an Inbox where new mail arrives. Users can also create additional *private folders* or *mailboxes,* where they file and store messages they have read. Folders can contain other folders in a hierarchical tree.

In addition to the Inbox and private folders, users can also own *shared folders.* A shared folder is similar to a email group, but instead of messages going into each members inbox, messages addressed to the shared folder go into a private folder in each users message store.

For example, the network design team of the Stream Corporation have created a shared folder called `network-design@bridge.net`. Messages sent to this address go into private shared folders called `network-design` in each members message store space. Each member brings up that mailbox in order to see the mail sent to that address. The client syntax for accessing the mailbox is `#shared/<folder name>`. In this example members would enter `#shared/network-design` to bring up the shared folder. On recent IMAP clients that support namespace, these folders are visible automatically.

Members can read then delete messages from their *view* of the shared folder, but the actual messages remain in the mailbox of shared folder's *owner*. The owner of the shared folder is designated when the shared folder is first created. The owner is the only member of the group who can permanently expunge the message from the shared folder. The designated owner of the shared folder uses the syntax `#shared.<folder name>/Inbox` to access the shared folder. (In our example that would be `#shared.network-design/Inbox`.)

Sun OpenWindows Mail Tool V3 Format Conversion

Before a message can be processed by the Sun Message Store, the body of the message must be in Multipurpose Internet Mail Extensions (MIME) format. (The Sun Message Store stores messages in MIME format only.) When a message enters the Sun Message Store queue, it checks the message body format. If the message body is in MIME format, the Sun Message Store automatically processes the message. If the message body is in Sun OpenWindowsMail Tool V3 format (the message was generated in Mail Tool), the Sun Message Store converts the message body format to MIME. This feature is not user configurable.

Simultaneous Connections

SIMS allows simultaneous IMAP and POP client connections. While this is standard for most IMAP clients, for POP clients this means that if a connection is dropped, or a user wants more than one connection, then a user can immediately reconnect and have access to his messages without having to wait for the POP server to time-out.

Simultaneous IMAP Connections

When multiple IMAP clients connect to a mailbox, all flag changes (new, seen, deleted, answered, and so on) and message states (new mail, removed message) are shared between clients by means of the IMAP protocol.

Simultaneous IMAP and POP Connections

When IMAP and POP clients connect to a mailbox, all flag changes (new, seen, deleted, answered, and so on) and message states (new mail, remove message) are shared between IMAP clients. POP clients, however, will not see IMAP deleted messages. When a POP client connects to a mailbox, it grabs a snapshot of the mailbox which persists until the POP session is terminated.

Simultaneous POP Connections

When multiple POP clients connect to a mailbox, each grabs a unique snapshot of the mailbox state at connect time. Each connection’s unique snapshot persists until the POP session is terminated. Any changes to a mailbox’s state are seen by clients that connect after these changes are made.

APOP authentication for POP3

APOP is a POP command that the mail client can use as an alternative to USER/ PASS (RFC 1939) for secure authentication. Unlike USER/PASS, APOP does not use the user’s plaintext password for authentication. Instead, it uses an encoding of the password together with a challenge string. This encoding uses the md5 method.

Message Access Protocols

This section provides further explanation of the message access protocols, including:

- Message access protocol transparency
- Security between mail client and mail server

Message Access Protocol Transparency

The Sun Message Store and `/var/mail` support the Internet Mail Access Protocol version 4 (IMAP4) and the Post Office Protocol version 3 (POP3). Therefore, a mail client user can access either message store using either protocol. For example, a user's Inbox and other folders can be stored on the Sun Message Store and accessed using IMAP4 or POP3 or they can be stored in `/var/mail` and accessed using IMAP4 or POP3.

Message Access Service Specifications

Message access services refers to the protocol servers, software drivers, and libraries that support client access to the message store. The key to this component is the Internet Message Access Protocol version 4 (IMAP4), implemented via the c-client Mail API library. This component is also responsible for the Post Office Protocol version 3 (POP3). The key features of this component are:

- **IMAP4 Revision 1**—This protocol has been extended to optimize network usage and improve low-bandwidth performance.
- **Advanced POP3**—In addition to the mandatory POP3 command set, message access services supports the optional `TOP` and `UIDL` commands. The `UIDL` implementation is based on IMAP4 Universal Identifiers.
- **Orthogonal Message Store Access**—Multiple access protocols (IMAP and POP3) are supported from a common message store. Similarly, both message stores (Solaris Mailbox Format and Sun Message Store) support both access protocols.
- **Support for Sun Mail tool V3 Attachment Format**—Documents received in Sun V3 format are automatically converted in the message stores to MIME.

CHAPTER 10

SIMS Adminstration Console

SIMS Administration Console is a GUI-based tool that allows administering and maintaining SIMS administrative tasks, including domain hosting.

Topics in this chapter include:

- SIMS administration services
- SIMS administration console
- Administration server
- Mail server authorities
- Administration services

SIMS Administration Services

SIMS administrative services enable you to fine-tune the default configuration, maintain, monitor, and troubleshoot the SIMS components. It is composed of the following elements:

- Administration console
- Administration server
- SIMS components
- Remote Method Invocation (RMI)

FIGURE 10-1 illustrates these elements. The following sections provide further explanation.

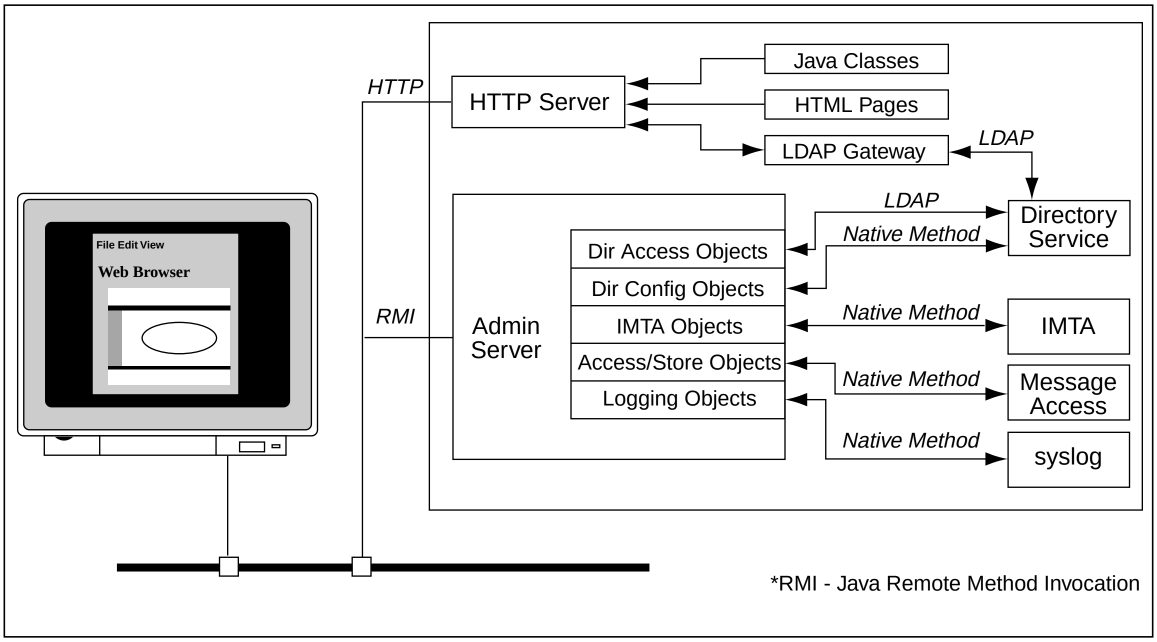


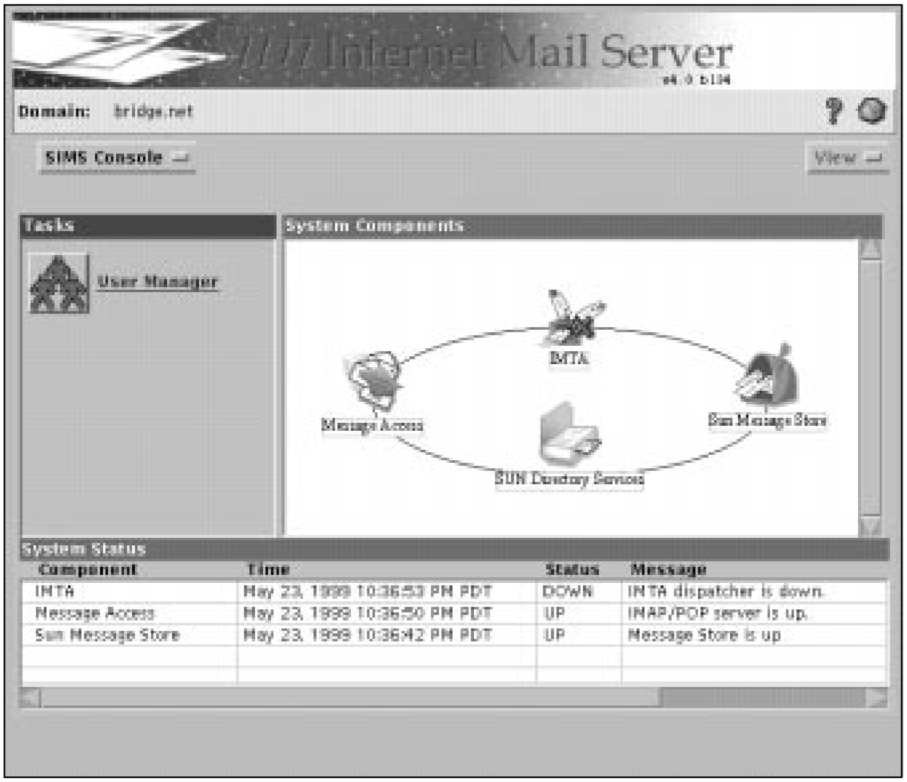
FIGURE 10-1 Sun Internet Mail Server Administrative Architecture

You can administer SIMS from the same machine on which all SIMS components are installed, or if desired, remotely from any machine on the network. The remote machine must be able to run the Netscape browser 4.06 and higher. Other browsers or other versions of HotJava may not work with the Administration Console.

SIMS Administration Console

The SIMS Administration Console provides the graphical user interface that enables you to configure, maintain, monitor, and troubleshoot the SIMS components, including domain hosting capabilities. The SIMS Administration Console runs on either the Netscape or the HotJava browser. FIGURE 10-2 shows a sample SIMS Administration Console page.

Note – Only one administrator can be logged on to the Administration Console at any given time.



System Status

Component	Time	Status	Message
IMTA	May 23, 1999 10:36:53 PM PDT	DOWN	IMTA dispatcher is down.
Message Access	May 23, 1999 10:36:50 PM PDT	UP	IMAP/POP server is up.
Sun Message Store	May 23, 1999 10:36:42 PM PDT	UP	Message Store is up

FIGURE 10-2 Sample Administration Console Page

The applet contains interfaces to the managed objects that reside in the Administration Server.

See Chapter 11, “Delegated Management Administration,” for summaries of tools that different types of administrators could use to perform domain hosting administrative tasks.

See Chapter 11, “SIMS Configuration Files” in the The *Sun Internet Mail Server 4.0 Installation Guide* for a list of configuration files associated with each component of SIMS.

SIMS Administration Server

SIMS Administration server contains two relevant elements:

- Managed objects
- HyperText Transfer Protocol (HTTP) server

A managed object is a collection of configurable attributes, for example, a collection of attributes for the directory service. (Note that a managed object does not necessarily map to a SIMS component.)

An HTTP server resides on the Administration Server to provide bootstrap capabilities for certain Java elements. A HyperText Markup Language (HTML) file causes an initial applet and managed objects to be loaded from the Administration Server to the browser. After the initial applet takes control, it uses its managed object interfaces to communicate with the Administration Server.

You do not need to configure or interact in any way with the Administration Server.

Communication with SIMS Components

Each SIMS component except the directory service (specifically, the directory access object that manages the user and group entries) communicates with the Administration Server using that component’s native format. The directory access object interfaces with the Administration Server using the Lightweight Directory Access Protocol (LDAP, The directory configuration object that manages the configuration of the directory service itself interfaces with the Administration Server using the native format.)

RMI

The Java Remote Method Invocation (RMI) enables the Administration Server and Console, which are running in either different address spaces on the same machine or on different machines, to communicate with each other. RMI enables the remote managed objects that reside on the Administration Server to be manipulated by the managed object interfaces that reside in the Admin Console.

Mail Server Authorities

The Sun Internet Mail Server features two levels of security: one level from the directory service and another from the UNIX file system. Specifically, you must have access permission from the following authorities.

Directory Services

Directory services requires user name and encrypted password to access the Administration Console.

UNIX File System

`inetmail` (owner of Sun Internet message store and configuration files)—Requires `root` permission to invoke SIMS utilities and commands from the command line.

Solaris format mailbox files (owned by individual users)—Requires either ownership of the mailbox to be manipulated or `root` permission.

Note – When a client machine binds to a directory server, the password used in the bind request is passed in clear text rather than encrypted text.

Administration Services

The administration services component contains the client and server software responsible for managing all the components of the SIMS. It also provides the GUI for some system functionality.

Key features of this component are:

- Installation
- Initialization and setup
- Configuration
- Maintenance
- Error recovery

Component Architecture

The administration services component manages:

- Sun directory services
- IMTA
- Sun message store
- Security

The following submodules provide the core functions of the Administration services:

- Browser-based Java administration console
- Downloadable Java applets
- Java-based administration server
- HTTP server

Who is the SIMS Administrator?

The SIMS Administrator has the right to perform the following tasks:

- Login to the SIMS administration console
- Change configuration from the SIMS Administration Console
- Change the directory content both from the Console and from command line.

More than one SIMS administrator can exist. A user can become a SIMS administrator by using the CLI `imadmin add admin` commands. However, only one administrator can log into the SIMS Administration Console at any time.

During the installation, one administrator is created. The default name is `siteadmin`, the default password is the password for the UNIX user `inetmail`.

Note – The password is encoded from the admin console machine to the admin server machine, but when the server machine binds to the directory server, the password used in the bind request is passed in clear rather than encrypted text.

Changing Configuration Files

In general it is not recommended that you edit the configuration files in any way. Use the SIMS Administration Console and the CLI commands to manipulate the configurations. In some situations, especially for IMTA, not all of the parameters that are supported in the configuration file are configurable from the Administration Console. If there's a need to manually edit the configuration file, you should use the `imedit` tool that is provided with SIMS4.0 to edit the configuration file.

CHAPTER 11

Delegated Management Administration

This chapter provides you with a summary of the types of administrators, the tasks they can perform, and the tools that they can use to perform the delegated management tasks on domains that are hosted by service providers (SP)s.

Topics in this chapter include:

- Administrators' tasks and tools
- Delegated management features

Administrators’ Tasks and Tools

SIMS supports four different types of administrators to perform domain hosting tasks at the domain level. TABLE 11-1 shows a list of different types of domain-level tasks that each of the four types of administrators could perform along with the tools that are available to these administrators.

TABLE 11-1 Summary of Administrator’s Tasks and Tools

Type of Administrator	Site	Allowable Task	Using Tool
SIMS Administrator	SP	Create hosted domains	Administration CLIs
			Administration Console
		Create a delegated administrator	Administration CLIs
	Subscriber	Manage hosted domains	Administration CLIs
		Manage user and groups	Administration Console
			Administration CLIs
Delegated Administrator	Subscriber	Manage users and distribution lists	Delegated Management Console
			Delegated Management CLIs
		Personal change preferences	Delegated Management Console
		Bulk-loading new users and distribution lists	Delegated Management CLIs
Distribution List Owner	Subscriber	Manage distribution lists	Delegated Management Console
End-User	Subscriber	Change personal preferences	Delegated Management Console

TABLE 11-3 shows a complete list of delegated management tasks that can be performed by each of the four types of administrators.

See Chapter 6, “Domain Hosting with SIMS” for domain hosting feature components and specifications, including the Delegated Management Console and Domain Management server.

See Chapter 10, “SIMS Administration Console” for information about the architecture of the Console and the types of services that it offers, including domain hosting capabilities.

See Chapter 11, “SIMS Configuration Files” in the The *Sun Internet Mail Server 4.0 Installation Guide* for a list of configuration files that are used by the SIMS components.

See Chapter 17, “Domain Hosting”, in the *Sun Internet Mail Server 4.0 Administrator’s Guide* for instruction on creating domain, administrator, and customization.

See the *Sun Internet Mail Server 4.0 Delegated Management Guide* for instructions on how a delegated administrator for a hosted domain can perform tasks for end-users and distribution lists.

TABLE 11-2 shows the names and types of the utilities that the administrators could use to perform their domain-level tasks.

TABLE 11-2 SIMS Domain Hosting Tools

Type of Tool	Type of Utility	Name of Utility	Function
SIMS Administration CLIs (Site and Subscriber)	Administrative	imadmin	Provisioning Channel configuration
	Monitoring	immonitor	System monitoring
Administration Console	Administrative (limited)		
Delegated Management CLIs	Administrative	imadmin	Provisioning

Delegated Management Tasks

TABLE 11-3 shows the types of tools that the administrator can use, which is used based on the domain hosting functionality.

TABLE 11-3 SIMS 4.0 Delegated Management Tasks

Task	Type of Administrator/User	Type of Tool
Domain Creation and Management		
Create domains	SIMS administrator	Administration CLIs
		Administration Console
Modify and delete domains	SIMS administrator	Administration CLIs
Delegated Administration Setup		
Create a Delegated Administrator for each hosted domain	SIMS administrator	Administration CLIs
Set and modify user-level restrictions		
User Creation and Management		
Bulk-load domain-level users	SIMS administrator	Administration CLIs
Create, modify, and delete users for each hosted domain		Administration Console
	delegated administrator	Administration CLIs
		Delegated Management Console
Modify user preferences	SIMS administrator	Administration CLIs
		Administration Console
	delegated administrator	Administration CLIs
		Delegated Management Console
	End-user	Delegated Management Console
Distribution List Management		
Create, modify distribution lists	SIMS administrator	Administration CLIs
		Administration Console
	delegated administrator	Administration CLIs

TABLE 11-3 SIMS 4.0 Delegated Management Tasks

Task	Type of Administrator/User	Type of Tool
		Delegated Management Console
	distribution list owner	Delegated Management Console
Modify distribution lists	distribution list owner	Delegated Management Console

Glossary

ACAP Application Configuration Access Protocol. A protocol which enhances IMAP by allowing the user to set up address books, user options, and other data for universal access.

access control rules Rules specifying user permissions for a given set of directory entries or attributes.

access control list (ACL) A set of data associated with a directory that defines the permissions that users and/or groups have for accessing it.

Administration Console or Admin Console A GUI (graphical user interface) which enables you to configure, monitor, maintain, and troubleshoot the SIMS components.

address mapping See forward address mapping or reverse address mapping.

address token The address element of a rewrite rule pattern.

Administration Services A service daemon that administers components of SIMS through a GUI.

agent In the client-server model, the part of the system that performs information preparation and exchange on behalf of a client or server application. See also *MTA*.

alias An alternate name of an email address.

alias file A file used to set aliases not set in a directory, such as the postmaster alias.

APOP Authenticated Post Office Protocol. Similar to the Post Office Protocol (POP), but instead of using a plaintext password for authentication, it uses an encoding of the password together with a challenge string.

attribute The form of information stored and retrieved by the directory service. Directory information consists of entries, each containing one or more attributes. Each attribute consists of a type identifier together with one or more values. Each directory read operation can retrieve some or all attributes from a designated entry.

attribute index An index, or list, of entries which contains a given attribute or attribute value.

autoreply option file A file used for setting options for autoreply, such as vacation notices.

backbone The primary connectivity mechanism of a distributed system. All systems that have connectivity to an intermediate system on the backbone are connected to each other. This does not prevent you from setting up systems to bypass the backbone for reasons of cost, performance, or security.

bang path An address for sending e-mail via UUCP that specifies the entire route to the destination computer. It separates each host name with an exclamation point, which is also known as a bang. For example, the bang path `midearth!shire!bilbo!jsmith` would go to the `jsmith` user account on the `bilbo` host, which is reached by first going to `midearth` and then `shire`.

CA Certificate Authority. An organization that issues digital certificates (digital identification) and makes its public key widely available to its intended audience.

directory cache A temporary storage of information that has been retrieved from the directory.

Certificate Authority See CA.

channel An interface with another SIMS component, another email system, or a mail user agent.

character set labels A name or label for a character set.

client-server model A computing model in which powerful networked computers provide specific services to other client computers. Examples include the name-server/name-resolver paradigm of the DNS and fileserver/file-client relationships such as NFS and diskless hosts.

composition The process of constructing a message by the Mail User Agent (MUA). See also *MUA*.

configuration file A file that contains the configuration parameters for a specific component of the SIMS system.

congestion thresholds A disk space limit that can be set by the system administrator that prevents the database from becoming overloaded by restricting new operations when system resources are insufficient.

conversion channel Converts body of messages from one form to another.

cookie Cookies are text-only strings entered into the browser's memory automatically when you visit specific web sites. Cookies are programmed by the web page author. Users can either accept or deny cookies. Accepting the cookies allows the web page to load more quickly and is not a threat to the security of your machine.

ciphertext Text which has been encrypted. Opposite of plaintext.

daemon A UNIX program that is not invoked explicitly, but lies dormant waiting for some condition(s) to occur. The instigator of the condition need not be aware that a daemon is lurking (though often a program will commit an action only because it knows that it will implicitly invoke a daemon). Typical daemons are print spoolers, e-mail handlers, and schedulers that start up another process at a designated time or condition.

data store A store that contains directory information, typically for an entire directory information tree.

DC tree Domain Component tree. A directory information tree that mirrors the DNS network syntax. An example of a distinguished name in an DC tree would be `cn=billbob,dc=bridge,dc=net,o=internet`

defragmentation The Multiple Internet Mail Extensions (MIME) feature that enables a large message that has been broken down into smaller messages or fragments to be reassembled. A Message Partial Content-Type header field that appears in each of the fragments contains information that helps reassemble the fragments into one message. See also *fragmentation*.

delegated administrator A person who has the privileges to add, modify, delete, and search for group or user entries at a specified hosted domain.

Delegated Management Console A web browser-based software console that allows delegated administrators to add and modify users and groups to a hosted domain. Also allows end users to change their password, set message forwarding rules, set vacation rules, and list distribution list subscriptions.

delegated management server A daemon program that handles access control to the directory by hosted domains.

denial of service attack A situation where an individual intentionally or inadvertently overwhelms your mail server by flooding it with messages. Your server's throughput could be significantly impacted or the server itself could become overloaded and nonfunctional.

dereferencing an alias Specifying, in a bind or search operation, that a directory service translate an alias distinguished name to the actual distinguished name of an entry.

destination channel The last element of a host/domain rewrite rule, in whose queue a message should be placed in for delivery.

directory cache A cache containing the directory information used by the IMTA to deliver mail.

directory context The point in the directory tree information at which a search begins for entries used to authenticate a user and password for Sun Message Store access.

directory entry A set of directory attributes and their values identified by its distinguished name. Each entry contains an object class attribute that specifies the kind of object the entry describes and defines the set of attributes it contains. Also called the *IMTA directory cache.*

directory information tree The tree-like hierarchical structure in which directory entries are organized. Also called a DIT. DITs can be organized along the DNS (DC trees) or Open Systems Interconnect networks (OSI trees).

directory schema The set of rules that defines the data that can be stored in the directory.

directory service A logically centralized repository of information. The component in SIMS that stores user, distribution list, and configuration data.

directory synchronization Because information stored in the directory service is updated as new entries are added, modified and deleted, the information in the IMTA directory cache must be periodically updated with the current information in the directory service. This process is called directory synchronization. Sometimes called a dirsync in reference to the `imta dirsync` command.

dirsync option file A file used to set options for the `dirsync` program which cannot be set through the command line.

disconnected state The mail client connects to the server, makes a cache copy of selected messages, then disconnects from the server.

distinguished name The comma-separated sequence of attributes and values that specify the unique location of an entry within the directory information tree. Often abbreviated as DN.

distribution list A list of email addresses (users) that can be sent a message by specifying one email address. Also called a group. See also *expansion, member, moderator, owner,* and *alias*.

distribution list owner An individual who is responsible for a distribution list. An owner can add or delete distribution list members. See also *distribution list*, *expansion*, *member*, and *moderator.*

DIT See *directory information tree.*

DN Distinguished name.

DNS Domain Name System. A distributed name resolution software that allows computers to locate other computers on a UNIX network or the Internet by domain name. DNS servers provide a distributed, replicated, data query service for translating host names into Internet addresses.

DNS database A database of domain names (host names) and their corresponding IP addresses.

domain A group of computers whose host names share a common suffix, the *domain name*. Syntactically, an Internet domain name consists of a sequence of names (labels) separated by periods (dots), for example, `tundra.mpk.ca.us`.

domain quota The amount of space, configured by the system administrator, allocated to a domain for email messages.

domain rewriting rules See also *rewrite rules*.

domain template The part of a rewrite rule that defines how the host/domain portion of an address is rewritten. It can include either a full static host/domain address or a single field substitution string, or both.

dsservd A daemon that operates that accesses the database files that hold the directory information, and communicates with directory clients using the LDAP protocol.

EMAPI Extended MAPI Service Provider. Transparently turns Microsoft Exchange client into an Internet standard IMAP/LDAP client. See also *IMAP*, *LDAP*.

encryption Scrambling the contents of a message so that its contents cannot be read without the encryption, or code key.

entries User, group, or organizational data used to configure message accounts.

envelope The part of an Internet mail message that contains the delivery information. The envelope contains the originator and recipient information associated with a message.

ESMTP Extended Simple Mail Transfer Protocol. An Internet message transport protocol.

expander Part of an electronic mail delivery system which allows a message to be delivered to a list of addressees. Mail expanders are used to implement mailing lists. Users send messages to a single address (e.g., hacks@somehost.edu) and the mail expander takes care of delivery to the mailboxes in the list. Also called *mail exploders*.

expansion This term applies to the IMTA processing of distribution lists. The act of converting a message addressed to a distribution list into enough copies for each distribution list member.

expunge The act of marking a message for deletion and then permanently removing it from you INBOX.

external channel An interface between the IMTA and either another SIMS component or another component outside the SIMS email system.

failover The automatic transfer of a computer service from one system to another to provide redundant backup.

Filesharing Transport This type of transport moves messages between the UNIX operating system and the PC running a client through a shared file system available to both platforms. When a channel is configured to use filesharing transport, the shared directory to use for the file exchange must be specified.

firewall A dedicated gateway machine with special security precautions used to service outside network, especially Internet, connections and dial-in lines. The idea is to protect a cluster of more loosely administered machines hidden behind the firewall from unwanted entry from outside the firewall.

folder Named place where mail is stored. Also called a *mailbox*. Inbox is a folder that stores new mail. Users can also have folders where mail can be stored. A folder can contain other folders in a hierarchical tree. Folders owned by a user are called *private folders*. See also *shared folders*.

Folder Check A utility which checks the accessibility of messages and folders and verifies links. This utility is used as part of the regular maintenance of SIMS.

forward address mapping Message envelopes, TO:address, are processed to a mapping table. The result of the mapping is tested. If necessary, the exact form of the envelope is exchanged for another which can then be processed by a different, and perhaps non-compliant RFC 822, mail system.

FQDN See fully qualified domain name.

fragmentation The Multiple Internet Mail Extensions (MIME) feature that allows the breaking up of a large message into smaller messages. See also *defragmentation*.

full static host/domain address The portion of a host/domain address elements set off by decimals as part of the domain template. See also *domain template*.

fully qualified domain name The full name of a system, consisting of its local host name and its domain name. For example, *class* is a host name and *class.sun.edu* is an fully qualified domain name. A fully qualified domain name should be sufficient to determine a unique Internet address for any host on the Internet. The same naming scheme is also used for some hosts that are not on the Internet, but share the same name-space for electronic mail addressing. A host which does not have a fully qualified domain name must be addressed using a bang path.

gateway The terms *gateway* and *application gateway* refer to systems that do translation from one native format to another. Examples include X.400 to/from RFC 822 electronic mail gateways. A machine that connects two or more electronic mail systems (especially dissimilar mail systems on two different networks) and transfers messages between them. Sometimes the mapping and translation can be complex, and it generally requires a store-and-forward scheme whereby the message is received from one system completely before it is transmitted to the next system after suitable translations.

global log manager A utility that handles log information from each Sun Internet Mail Server component.

group Same as a distribution list.

group folders These contain folders for shared and group folders. See *shared folder.*

header The part of an Internet mail message that is composed of a field name followed by a colon and then a value. Headers include delivery information, summaries of contents, tracing, and MIME information.

hosted domain An email domain that is outsourced by an ISP. That is, the ISP provides email domain hosting for an organization by operating and maintaining the email services for that organization. A hosted domain shares the same SIMS host with other hosted domains. In earlier LDAP-based email systems, a domain was supported one or more email server hosts. With SIMS, many domains can be hosted on a single server. Hosted domains are also called *virtual hosted domains* or *virtual domains*.

host name The logical name assigned to a computer. On the Web, most hosts are named `www`; for example, `www.mycompany.com`. If a site is composed of several hosts, they might be given different names such as `support.mycompany.com` and `sales.mycompany.com`. `support` and `sales` are the host names, `mycompany` is the subdomain name, and `com` is the top-level domain name.

IMAP4 Internet Message Access Protocol. IMAP4 provides advanced disconnected mode client access.

IMTA Internet Message Transfer Agent. IMTA routes, transports, and delivers Internet Mail messages within the email system.

internal channel An interface between internal modules of the IMTA. Internal channels include the reprocessing, conversion, and defragmentation channels. These channels are not configurable.

Internet A collection of networks interconnected by a set of routers that allow them to function as the largest single world-wide virtual network.

internet protocol address A 32-bit address assigned to hosts using TCP/IP. Also called the *IP address* and *internet address.*

invalid user An error condition that occurs during message handling. When this occurs, the message store sends a communication to the Internet Message Transport Agent (IMTA), the message store deletes its copy of the message. The IMTA bounces the message back to the sender and deletes its copy of the message.

ISP Internet Service Provider. A company that provides internet services to its customers including email, electronic calendaring, access to the world wide web, and web hosting.

job controller An IMTA daemon responsible for scheduling message delivery. Job controller also controls channel queues and determines the order of processing. Requests are processed in the order in which they are received by the system.

knowledge information Part of the directory service infrastructure information. The directory server uses knowledge information to pass requests for information to other servers.

LDAP Lightweight Directory Access Protocol. LDAP is a protocol used for the storage, retrieval, and distribution of information, including user profiles, distribution lists, and configuration data.

LDAP referrals An LDAP entry that consists of a symbolic link (referral) to another LDAP entry. An LDAP referral consists of an LDAP host and a distinguished name. LDAP referrals are often used to reference existing LDAP data so that this data does not have to be replicated. They are also used to maintain compatibility for programs that depend on a particular entry that may have been moved.

LDAP Server A software server that maintains an LDAP directory and services queries to the directory. The Sun Directory Services and the Netscape Directory Services are implementations of an LDAP Server.

LDAP server failover A backup feature for LDAP servers. If one LDAP server fails, the system can switch over to another LDAP server.

LDAP filter A way of specifying a set of entries, based on the presence of a particular attribute or attribute value.

LDBM LDAP Data Base Manager.

LDIF LDAP Data Interchange Format. A data format used to represent LDAP entries in text form.

local channel A channel that allows you to determine delivery options of local users and delivers mail to Solaris Operating Environment mailboxes.

lookup Same as a search, using the specified parameters for sorting data.

mailbox A place where messages are stored and viewed. See *folder*.

managed object A collection of configurable attributes, for example, a collection of attributes for the directory service.

mapping tables Two column tables which transform, map, an input string into an output string.

master directory server The directory server that contains the data that will be replicated.

master message catalog Contains message catalogs for the SIMS components.

master program A channel program that initiates a message transfer to another interface on its own.

member A user or group who receives a copy of an email addressed to a distribution list. See also *distribution list*, *expansion*, *moderator*, and *owner.*

Message Access and Store The SIMS components which store user messages and allow for retrieval and processing of messages.

Message Access Services Consists of protocol servers, software drivers, and libraries which support client access to the message store.

message access services The drivers and libraries that support client access to the SIMS message store.

message catalogs The log messages, command line responses, and graphical user interface screen text contained in the SIMS components.

message submission The client Mail User Agent (MUA) transfers a message to the mail server and requests delivery.

MIB Management Information Base. A collection of objects that can be accessed via a network management protocol. See also *SMI.*

MIME Multipurpose Internet Mail Extensions. A format for defining email message content.

moderator A person who first receives all email addressed to a distribution list before A) forwarding the message to the distribution list, B) editing the message and then forwarding it to the distribution list, or C) not forwarding the message to the distribution list. See also *distribution list*, *expansion*, *member*, and *owner.*

MTA Message Transfer Agent. An OSI application process used to store and forward messages in the X.400 Message Handling System. Equivalent to Internet mail agent. See *IMTA*.

MUA Mail User Agent. The client applications invoked by end users to read, submit, and organize their electronic mail.

mx record Mail Exchange Record. A DNS resource record stating a host that can handle electronic mail for a particular domain.

name resolution The process of mapping an IP address to the corresponding name. See also *DNS*.

namespace The space from which an object name is derived and understood. Files are named within the file namespace, domain components are named within the domain namespace.

naming attribute The final attribute in a directory information tree distinguished name. See also *relative distinguished name.*

naming context A specific subtree of a directory information tree that is identified by its DN. In SIMS, specific types of directory information are stored in naming contexts. For example, a naming context which stores all entries for marketing employees in the XYZ Corporation at the Boston office might be called ou=mktg, ou=Boston, o=XYZ, c=US.

NIS A distributed network information service containing key information about the systems and the users on the network. The NIS database is stored on the master server and all the replica or slave servers.

NIS+ A distributed network information service containing hierarchical information about the systems and the users on the network. The NIS+ database is stored on the master server and all the replica servers.

nondelivery report During message transmission, if the IMTA does not find a match between the address pattern and a rewrite rule, the IMTA sends a nondelivery report back to the sender with the original message.

notary messages Text messages sent by the MTA to an email sender indicating delivery or non-delivery status of a sent message.

object class A template specifying the kind of object the entry describes and the set of attributes it contains. For example, SIMS specifies an `emailPerson` object class which has attributes such as `commonname`, `mail` (email address), `mailHost`, and `mailQuota`.

off-line state The mail client fetches messages from a server system to a client system, which may be a desktop or portable system and may delete them from the server. The mail client downloads the messages where they can be viewed and answered.

on-line state A state in which messages remain on the server and are remotely responded to by the mail client.

option files IMTA option files contain global parameters used to override default values of parameters which apply to IMTA as a whole, such as sizes for various tables into which various configuration and alias files are read.

OSI tree A directory information tree that mirrors the Open Systems Interconnect network syntax. An example of a distinguished name in an OSI tree would be `cn=billt,o=bridge,c=us`

permanent failure An error condition that occurs during message handling. When this occurs, the message store deletes its copy of an email message. The Internet Message Transport Agent (IMTA) bounces the message back to the sender and deletes its copy of the message.

pipe channel A channel which performs delivery of messages via a per-user-site-supplied program. These programs must be registered in SIMS by the system administrator, and thus do not pose a security risk.

plaintext Unencrypted readable text. The opposite of cypher text

plaintext authentication Authentication that occurs by sending passwords over the network in plaintext. Considered a security problem since plaintext passwords can be easily captured over a network.

POP Post Office Protocol. POP provides remote access support for older mail clients.

populating the directory Entering information for users and distribution lists to the SIMS directory service.

protocol A formal description of messages to be exchanged and rules to be followed for two or more systems to exchange information.

provisioning The process of adding, modifying or deleting entries in the SIMS directory service. These entries include users and groups.

provisioning commands SIMS commands that provide provisioning functions. These commands are prefaced with `imadmin`.

proxy The mechanism whereby one system “fronts for” another system in responding to protocol requests. Proxy systems are used in network management to avoid having to implement full protocol stacks in simple devices, such as modems.

public key encryption A cryptographic method that uses a two-part key (code) that is made up of public and private components. To encrypt messages, the published public keys of the recipients are used. To decrypt the messages, the recipients use their unpublished private keys known only to them.

purge The process of permanently removing messages that have been deleted and are no longer referenced in user and group folders and returning the space to the Sun Message Store file system. See also *backup* and *restore*.

quota See user quota.

referral A process by which the directory server returns an information request to the client that submitted it, with information about the Directory Service Agent (DSA) that the client should contact with the request. See also *knowledge information*.

relaying A message is passed from one mail server to another mail server.

relative distinguished name The final attribute and its value in the attribute and value sequence of the distinguished name. See also *distinguished name.*

replica directory server The directory that will receive a copy of all or part of the data.

reprocessing channel Performs deferred processing. The reprocessing channel is the intersection of all other channel programs. It performs only the operations that are shared with other channels.

restore The process of restoring the contents of folders from a backup device to the Sun Message Store. See also *backup* and *purge*.

reverse address mapping Addresses are processed to a mapping table, with a reversal database, generally substituting a generic address, possibly on a central machine, for an address on a remote or transitory system.

rewrite rules Also known as domain rewriting rules. A tool that the Internet Mail Transport Agent (IMTA) uses to route messages to the correct host for delivery. Rewrite rules perform the following functions: (1) extract the host/domain specification from an address of an incoming message, (2) match the host/domain specification with a rewrite rule pattern, (3) rewrite the host/domain specification based on the domain template, and (4) decide which IMTA channel queue the message should be placed in.

RFC Request For Comments. The document series, begun in 1969, describes the Internet suite of protocols and related experiments. Not all (in fact very few) RFCs describe Internet standards, but all Internet standards are published as RFCs. See `http://www.imc.org/rfcs.html`.

root entry The first entry of the directory information tree (DIT) hierarchy.

router A system responsible for determining which of several paths network traffic will follow. It uses a routing protocol to gain information about the network, and algorithms to choose the best route based on several criteria known as "routing metrics." In OSI terminology, a router is a Network Layer intermediate system. See also *gateway*.

routability scope Specifications which enable the IMTA to send messages by the most direct route, either to a specific user's folder, a group of folders, or to a mail host.

routing In an email system, the act of delivering a message based on addressing information extracted from the body of the message. The Internet Message Transfer Agent (IMTA) is the component responsible for routing messages.

safe file system A file system performs logging such that if a system crashes it is possible to rollback the data to a pre-crash state and restore all data. An example of a safe file system is Veritas File System, VxFS.

schema A set of rules which sets the parameters of the data stored in a directory. It defines the type of entries, their structure and their syntax.

sendmail This program acts as a mail transport agent for Solaris software. It is responsible for routing mail and resolution of mail addresses.

shared folder or shared mailbox *A* mailbox that can be viewed by members of a *distribution list*. Shared folders have an *owner* who can add or delete members to the group and can delete messages from a the shared folder. The can also have a moderator who can edit, block, or forward incoming messages.

SIMS administrator An individual who has a valid log in and password for the SIMS Admin Console. This person can also use this log in and password to execute the provisioning CLIs.

single field substitution string Part of the domain template that dynamically rewrites the specified address token of the host/domain address. See also *domain template*.

SKIP Simple Key management for IP. A security system that encrypts or scrambles the text of a message so only the receiving mail client or message server can decrypt or unscramble the text.

slave program A channel program that accepts transfers initiated by another interface.

smart host The mail server in a domain to which other mail servers, forward messages if they do not recognize the recipients.

SMTP Simple Mail Transfer Protocol. The Internet electronic mail protocol. Defined in RFC 821, with associated message format descriptions in RFC 822.

SMTP Dispatcher A multithreaded connection dispatching agent which allows multiple multithreaded servers to share responsibility for a given service, thus allowing several multithreaded SMTP servers to run concurrently and handle one or more active connections.

SMTP intranet or internet channel A channel dedicated to relaying messages between the IMTA and a group of SMTP hosts within, or outside of, your mail network.

SMTP router channel SMTP channel that handles messages between the IMTA and firewall host.

SNMP Simple Network Management Protocol. The network management protocol of choice for TCP/IP-based internets.

subordinate reference The naming context that is a child of the naming context held by your directory server. See also *knowledge information*.

Sun Directory Services Sun Microsystems' implementation of an LDAP directory server. Provides storage of, and access to, user profiles, distribution lists, and other SIMS information. The Sun Directory Services is one of the three main SIMS components along with the IMTA and MS/MA.

Sun Internet Mail Server An enterprise-wide, open-standards based, scalable electronic message-handling system.

Sun Message Store The server from which mail clients retrieve and submit messages.

SSL Secure Sockets Layer is an open, non-proprietary security protocol for authenticated and encrypted communication between clients and servers.

synchronization The update of data by a master directory server to a replica directory server.

table lookup With a table consisting of two columns of data, an input string is compared with the data within the table and transformed to an output string.

tailor file An option file used to set the location of various IMTA components.

transient failure An error condition that occurs during message handling. The remote Internet Message Transport Agent (IMTA) is unable to handle the message when it's delivered, but may be able to later. The local IMTA returns the message to the channel queue and schedules it for retransmission at a later time.

transport protocols Provides the means to transfer messages between message stores.

uid User identification. A unique string identifying a user to a system. Also referred to as a userid.

unsafe file system A file system that does not perform logging. If the system crashes, the state cannot be recreated and some data may be lost. You must also perform `imcheck` before activating message access to these files.

upper reference Indicates the directory server that holds the naming context above your directory server's naming context in the directory information tree (DIT).

user entry or user profile Fields that describe information about each user, required and optional, examples are: distinguished name, full name, title, telephone number, pager number, login name, password, home directory, etc.

user folders A user's email mailboxes.

user quota The amount of space, configured by the system administrator, allocated to a user for email messages.

user redirection The remote Internet Message Transport Agent (IMTA) cannot accept mail for the recipient, but can reroute the mail to a mail server that can accept it.

UUCP UNIX to UNIX Copy Program. A protocol used for communication between consenting UNIX systems.

valid user A condition that occurs during message handling. After the message store sends a communication to the Internet Message Transport Agent (IMTA), the IMTA deletes its copy of the message and it is now the message store's responsibility.

`/var/mail` The UNIX version 7 "`From`" delimited mailbox as implemented in the Solaris operating system.

virtual hosted domains or virtual domains See *hosted domains*.

workgroup Local workgroup environment, where the server performs its own routing and delivery within a local office or workgroup. Interdepartmental mail is routed to a backbone server. See also *backbone*.

X.400 A message handling system standard.

Index