Escaping Vendor Lock-in: Life After Microsoft Exchange

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

Part No: 819–5104–10
December 2005

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more U.S. patents or pending patent applications in the U.S. and in other countries.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, the Solaris logo, the Java Coffee Cup logo, docs.sun.com, Java, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this publication are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical or biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

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

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs brevets américains ou des applications de brevet en attente aux Etats-Unis et dans d'autres pays.

Cette distribution peut comprendre des composants développés par des tierces personnes.

Certaines composants de ce produit peuvent être dérivées du logiciel Berkeley BSD, licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays; elle est licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, le logo Solaris, le logo Java Coffee Cup, docs.sun.com, Java et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui, en outre, se conforment aux licences écrites de Sun.

Les produits qui font l'objet de cette publication et les informations qu'il contient sont régis par la legislation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes chimiques ou biologiques ou pour le nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des Etats-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la legislation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement designés, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.

Adobe PostScript™

051209@13215

Contents

Escaping Vendor Lock-in: Life After Microsoft Exchange 5

Revision History 6

Overview of the Problem 6

What Is the Sun Solution? 7

Overview of the Sun Java System Communications Services Architecture 8

Benefits of Sun Java System Communications Services 10

- Software Scalability 10
- Server Centralization 11
- Robust Security 11
- High Availability 12
- Open and Flexible Architecture 13
- Lower TCO 13
- The Next Step 14

Understanding the Sun Groupware Migration Toolkit and Methodology 14

- Migration and Coexistence Principles and Design 14
- What Factors Impact Migration? 20
- Addressing Typical Migration Concerns 21

Organizing the Migration Project 25

- Migration Problem Statement 25
- Migration Phases 27

For More Information 31

A Appendix 33

About Message Storage Concepts 33

Escaping Vendor Lock-in: Life After Microsoft Exchange

This document provides information for system architects and project managers interested in understanding:

- What is involved in migrating from Microsoft Exchange Server to Sun Java™ System Communications Services
- How to develop a co-existence strategy for Microsoft Exchange Server and Sun Java System Communications Services
- The architecture of the Sun Java System Communications Services platform
- How to organize a migration project

The components discussed in this document are:

- Sun Java System Communications Services 6, including:
  - Sun Java System Calendar Server
  - Sun Java System Communications Express
  - Sun Java System Communications Sync
  - Sun Java System Connector for Microsoft Outlook
  - Sun Java System Directory Server
  - Sun Java System Instant Messaging
  - Sun Java System Messaging Server
- Microsoft Exchange Server 5.5

Note – While this document focuses primarily on Microsoft Exchange 5.5, the Sun Groupware Migration Toolkit has already successfully been used with Microsoft Exchange 2000 and Microsoft Exchange 2003 running on Microsoft Windows 2000 and Microsoft Windows 2003. In addition, the toolkit has been used with Critical Path messaging software (frequently used in the telecommunications market).

Revision History

The following table shows the revision history of this document.

TABLE 1 Revision History

Date	Description of Changes
12/09/05	Initial release of this document.

Overview of the Problem

If you are managing an enterprise running Microsoft Exchange 5.5 on Windows NT 4.0, you find yourself in an especially uncomfortable and challenging position. Not only must you cut back on IT spending and do more with less, but you are also faced with the expiration of mainstream Exchange 5.5 support (as of December 31, 2003). In addition, Microsoft has discontinued support for Windows NT4 since the end of 2004, forcing an upgrade to Microsoft Windows 2000. In short, you need to make a critical decision about your new communication platform.

You could migrate your enterprise from Microsoft Exchange 5.5 to Microsoft Exchange 2003, but this is a complicated and costly endeavor. Newer versions of Microsoft Exchange run only on Microsoft Windows Server 2000 or 2003, and require the use of Microsoft's proprietary Active Directory. Thus, if you upgrade Microsoft Exchange, you need to upgrade the operating system, and you need to migrate the Exchange 5.5. directory to Active Directory as well. In-place upgrades are not possible, so the newer versions of Microsoft Exchange must be set up on separate servers and connected to the 5.5 environment to relocate mailboxes and databases to the new boxes.

The migration is costly, partly due to Microsoft's high software costs. A July 2004 report from Ferris Research estimates that a Microsoft Exchange 5.5 to Microsoft Exchange 2003 migration costs upwards of $250 per user.[1]

Perhaps most importantly, even if you decide to migrate from Microsoft Exchange 5.5 to a newer version of Microsoft Exchange, your enterprise might still face many of the weaknesses and limitations inherent in the Microsoft Exchange platform. Exploring alternative solutions is a must, particularly in a difficult economic environment.

[1] *The Cost of Migrating From Exchange 5.5 to Exchange 2003*, Ferris Research, July 2004

What Is the Sun Solution?

The Sun Microsystems solution to the problem just described is to exchange Microsoft Exchange on the server side for the Sun Java System Communications Services platform. The Sun platform includes the following components:

- Sun Java System Calendar Server
- Sun Java System Communications Express
- Sun Java System Communications Sync
- Sun Java System Connector for Microsoft Outlook
- Sun Java System Directory Server
- Sun Java System Instant Messaging
- Sun Java System Messaging Server

Together, the Messaging Server, Calendar Server, and Instant Messaging products provide robust, scalable, secure, and integrated email, calendaring, and instant messaging functionality. Furthermore, the Connector for Microsoft Outlook component seamlessly connects the Sun server software to Microsoft Outlook desktop clients, so that employees can continue to use Microsoft Outlook on their desktops. This avoids the painful and expensive exercise of retraining users on a new client. In addition, Communications Sync enables users to synchronize data from Calendar Server to a wide range of calendar clients and devices.

The end result is dramatic server consolidation, significantly reduced total cost of ownership (TCO), and a future-proofed investment in standards-based communication infrastructure. At the same time, Sun Java System Communications Services software delivers the necessary scalability, performance, and reliability, as well as the rich feature set and client support, required to maintain the services and experiences that end users demand.

Tip – For an overview of Sun Java System Communications Services component products, see Chapter 1, “Introduction to Deploying Communications Services,” in *Sun Java System Communications Services 6 2005Q4 Deployment Planning Guide*.

To implement the migration, the Sun Enterprise Messaging Consolidation Solution (EMCS) has been designed to enable enterprises to migrate from messaging platforms, such as Microsoft Exchange, to the Sun Java System Communications Services platform, and to do so *quickly* and *safely*. In addition to the set of Sun Java System Communications Services products previously mentioned, EMCS provides the following services:

- Sun Enterprise Messaging Reference Architecture
- Sun Groupware Migration Toolkit and Methodology
- Support and managed services

- Training

The following section discusses the architecture of the Sun solution in more detail.

Overview of the Sun Java System Communications Services Architecture

Sun Java System Communications Services uses a two-tiered messaging architecture that provides the optimum design for scalability and reliability. Instead of having a single host run all the components of a messaging system, a two-tiered architecture separates the components onto different machines. These separate components perform specific specialized functions. As the load for a particular functional component increases—for example, more Message Storage is required, or more outbound relaying is needed—you can add more servers to handle the larger loads.

The two-tiered architecture consists of an *access layer* and a *data layer*. The access layer is the portion of the architecture that handles delivery, message access, user login, and authentication. The data layer is the portion of the architecture that holds all the data. This includes the LDAP master servers and Messaging Server machines that are configured to store user messages.

To promote understanding of the Sun Java System Communications Services architecture, Sun created the Sun Enterprise Messaging Reference Architecture (SEMRA), which simulates a corporate Microsoft Exchange site consisting of 5,000 users. This reference architecture proved the concept that migration from Microsoft Exchange to a new Java Enterprise System target deployment works from a functional point of view.

Figure 1 illustrates the SEMRA logical architecture.

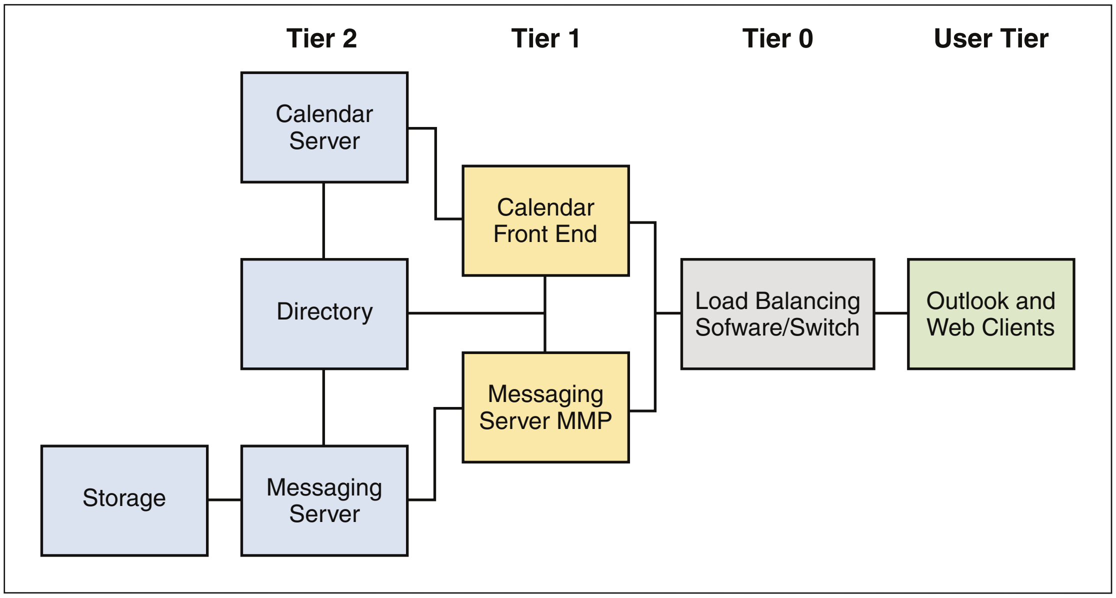

FIGURE 1 SEMRA Logical Architecture

In this figure, Tiers 0 through 2 represent the logical architecture for the SEMRA. The User Tier represents communications services users who utilize Microsoft Outlook or web clients.

In this architecture:

- Tier 0 contains the load balancer (software or switch).
- Tier 1 acts as the front end that provides access to the messaging applications.
- Tier 2 forms the layer where the messaging data is handled by the applications.

You can obtain more information about the Sun Enterprise Messaging Reference Architecture here:

`http://www.sun.com/service/refarch/collateral/enterprise_messaging_factsheet.pdf`

Benefits of Sun Java System Communications Services

Reasons to exchange a Microsoft Exchange deployment for Sun Java System Communications Services are numerous, and include the following:

- Software Scalability
- Server Centralization
- Robust Security
- High Availability
- Open and Flexible Architecture
- Lower TCO

The following sections discuss these reasons in more detail.

Software Scalability

Sun Java System Communications Services software is highly scalable. A March 2003 report from The Radicati Group[2] stated that the average Sun enterprise deployment had more than 5000 users on each server; in contrast, Microsoft Exchange 5.5 deployments averaged just 477 users per server.

Sun Java System Communications Services software scales well due to its robust messaging component technology. The message store is optimized and scales exceptionally well. The message transfer agent (MTA) is arguably one of the most robust, feature-rich MTAs on the market, with over 20 years in Internet mail deployments. Its multithreaded design is optimized to enable maximum message throughput, making it ideal for mass mailing, rich content delivery, and unified communication services. Indeed, Sun Java System Communications Services are embedded in most of the voice mail, unified messaging, and multi-media messaging providers available today. The component technology and architecture of Sun Java System Communications Services software enables the quality of service expected by both IT administrators and end users.

Microsoft Exchange, designed to scale to the workgroup level, not the enterprise level, does not scale nearly as well as Sun software. This limited scalability prevents Microsoft Exchange from providing a compelling TCO story. Managing dozens or hundreds of Microsoft Exchange servers is cumbersome and leads to high software, hardware, and administrative costs.

[2] *Messaging Total Cost of Ownership 2003 in Enterprise and Service Provider Environments*, The Radicati Group, March 2003

Server Centralization

Physically consolidating IT assets makes it easier to administer and maintain them, requiring fewer IT administrators and lowering labor costs. As Sun Java System Communications Services software is highly scalable, only a handful of Sun servers are required for even the largest enterprise deployments. Perhaps more importantly, these servers can also be centralized into one or two data centers, driving down labor costs while increasing server utilization. Sun Java System Communications Services software can be deployed in a centralized manner because the bandwidth-efficient communications protocols (including IMAP and SMTP) that are used between the Sun server software and the desktop client enable the server to physically reside far from the client.

In contrast, Microsoft Exchange cannot be centralized nearly as well as Sun Java System Communications Services software. This is largely related to the MAPI-based architecture used to connect Exchange and Outlook and its reliance on the Remote Procedure Call (RPC) networking protocol for client-server connections. An RPC-based solution, requiring multiple servers due to the heavy nature of the protocol, doesn't perform as well as IMAP and SMTP, which are used by Sun Java System Communications Services. With Microsoft Exchange, you end up with a highly distributed communications platform consisting of scattered islands of messaging data. This imposes heavy synchronization requirements, especially for public mail folders, calendar folders, and so forth.

Microsoft even recommends deploying additional Microsoft Exchange "cache" machines, especially for mail traffic, which consumes a great deal of network bandwidth. On the contrary, an IMAP-based product like Sun Java System Messaging Server benefits from using an IMAP design, which was, since its beginning, optimized for bandwidth efficiency. This explains why IMAP is so important for the telecommunication and service provider markets. These markets, whose third generation networks enable client devices direct IP access to mobile operator's infrastructures, depend upon IMAP. These markets also are counting on extensions to the protocol, such as those provided by the Lemonade IETF working group, to provide new services in the future.

Robust Security

Sun Java System Communications Services software provides robust security measures to protect the integrity of your data and the privacy of customers, partners, and employees. Sun Java System Communications Services supports Secure Sockets Layer (SSL) and Transport Layer Security (TLS) encryption to protect information assets. In addition, a messaging proxy can provide an additional layer of security at the firewall to further protect sensitive information. Extensive antispam and virus protection features also help protect the IT infrastructure and prevent lost productivity due to spam distraction or virus disruption.

Sun Java System Messaging Server supports the Mail Abuse Protection System (MAPS) Real-time Black Hole List to prevent email from known spammers, address verification to help ensure that messages are sent from valid domains, and relay blocking to prevent use of the server as a spam relay. Support for server-side rules enables system administrators and end users to configure filters on the server (before a message arrives on the desktop) to remove suspected spam, viruses, or other inappropriate content. Sun Java System Messaging Server has been preintegrated with Symantec Brightmail anti-spam filtering technology and SpamAssassin, an open source spam filtering software package. A conversion channel facilitates integration with other third-party, content-filtering software for richer spam and virus protection. In addition, a throttling mechanism helps prevent denial-of-service attacks.

In contrast to Sun, Microsoft products, including Microsoft Exchange, have consistently been plagued with notable security weaknesses. The distributed nature of the Microsoft Exchange solution means that multiple copies of files are stored on numerous servers, making it difficult to implement robust security measures. In the event of a virus attack, IT administrators literally have to shut down, clean up, and often, install security patches on each Microsoft Exchange server, one at a time. This process is slow, labor-intensive, and costly, especially when dealing with dozens or hundreds of servers. The bottom-line impact is higher IT labor costs and decreased worker productivity due to system downtime

High Availability

Sun Java System Communications Services software is highly reliable. Initially built for the service provider market, the software was designed from day one to meet the strict uptime requirements demanded by service providers and their millions of users. With more than 220 million mail boxes and 100 million calendar seats sold to service providers, enterprises, and other organizations, the Sun Java System Communications Services platform has been battle-tested in the real world, and has stood up to the test. Sun Java System Communications Services software integrates with Sun Cluster high-availability clustering software, to deliver virtually continual availability and rapid recovery, even if hardware failure does occur.

Microsoft Exchange cannot claim a similar story of reliability. Its complexity and typically distributed deployment often lead to system outages. Running multiple servers means an increased likelihood of downtime due to hardware problems. According to a March 2003 Radicati study[3] , Microsoft Exchange 5.5 users experienced unscheduled downtime more than twice as frequently as Sun users. The resulting dollar impact is an estimated annual downtime cost per user of $132 for Microsoft Exchange. Sun had a downtime cost of just $53 per user.

[3] *Messaging Total Cost of Ownership 2003 in Enterprise and Service Provider Environments*, The Radicati Group, March 2003

Open and Flexible Architecture

Sun Java System Communications Services software is flexible, open, and future-proof. It is based on open standards such as the Internet Message Access Protocol 4 (IMAP4), Post Office Protocol 3 (POP3), Simple Mail Transport Protocol (SMTP), Simple Network Management Protocol (SNMP), Short Message Service (SMS), and Lightweight Directory Access Protocol (LDAP). Sun's commitment to open standards is not just lip service. Several of the authors of these core Internet standards work for Sun and continue to contribute to the standards process.

Sun Java System Communications Services software also runs on a variety of operating systems, including the Solaris™ Operating System (SPARC® and x86 Platform) and Linux. In addition, it has well-documented APIs for extension and customization of messaging, calendaring, and instant messaging services. These published APIs and open standards make Sun Java System Communications Services software highly integratable with other enterprise applications, and facilitates migrations and upgrades of Sun Java System Communications Services products. In fact, the migration and upgrade costs of Sun Java System Communications Services software is less than 20 percent of that for Microsoft or IBM Lotus, according to a recent comparative study.[4]

Microsoft Exchange, like most other Microsoft products, is highly dependent on proprietary technology, such as MAPI. In addition, Microsoft is introducing new proprietary protocols in the short term, such as WebDAV. Microsoft Exchange leverages Microsoft's proprietary MAPI protocol and Active Directory, and only runs on Microsoft Windows. By committing to Microsoft Exchange, an enterprise is also committing itself to a Microsoft Windows-based operating system and developmental platform that might be difficult to integrate with non-Microsoft applications. Many corporate customers have grown weary of being locked into Microsoft's proprietary product strategy and the way it places them at the mercy of Microsoft's upgrade schedule. This schedule often involves a complicated and costly architectural overhaul every few years. Many Microsoft customers are seeking alternatives.

Lower TCO

High scalability, centralization, security and availability, combined with low software, hardware, and administrative costs, result in a compelling TCO for Sun Java System Communications Services software versus Microsoft Exchange. According to the Radicati Group's March 2003 comparative report on TCO in real-world enterprise deployments, Sun Java System Communications Services products delivered a three-year, average-loaded TCO of $214 per user, versus $439 per user for Microsoft Exchange and $407 for IBM Lotus. In today's environment where IT budgets are lean, TCO savings of this magnitude can have a meaningful and positive impact on the bottom line and will please both management and shareholders. And of course, these TCO savings will benefit the IT department as well; saved IT dollars can be allocated towards other IT initiatives.

[4] *Messaging Total Cost of Ownership 2003 in Enterprise and Service Provider Environments*, The Radicati Group, March 2003

The Next Step

Understanding the advantages of the Sun Java System Communications Services platform over a Microsoft Exchange environment is the first step in gaining an idea of what your future Sun platform will look like. Now that you have this understanding, the next step is to learn about the migration path itself, from Microsoft Exchange to your new target platform. This migration path complies to well-known generic requirements and also provides a thorough methodology that covers all migration aspects. In addition, Sun's migration approach enables you to define specific requirements that are reviewed and integrated as you accomplish your migration project.

Understanding the Sun Groupware Migration Toolkit and Methodology

To migrate from Microsoft Exchange 5.5 (and other groupware services) to the Sun Java System Communications Services platform, Sun provides you with the Sun Groupware Migration Toolkit and Methodology (SGMT). SGMT is not limited to the 5.5 version of Microsoft Exchange, nor to Exchange itself. In particular, SGMT currently covers MS Exchange 2000 and 2003, and Critical Path migration. To keep this document from becoming overly long, the discussion has been restricted to the Microsoft Exchange 5.5 use case.

Sun's methodology allows, by design, a safe migration project with no surprises, as long as it is fully developed. It covers step by step all solution aspects, not just the data migration per se. The methodology includes simulations and analysis before you attempt to run the actual migration. The methodology also enables you to implement several scenarios, in particular the online migration model with coexistence.

Migration and Coexistence Principles and Design

In the context of this document, the words *migration* and *coexistence* have precise meanings. Migration refers to the data migration of end users, in an offline or online mode, but can just as well refer to the project itself. Coexistence, which means you are running both Microsoft and Sun Java System platforms together, comes into play for an online migration. In this mode, you need coexistence of your system being migrated and the new system being migrated to. To summarize:

- **Offline Migration.** The source service is put offline and is no longer accessible by users. During this time (usually not more than one night or one weekend), the data is migrated from the source system to the target system.

- **Online Migration.** The source service is left online and the data migration occurs with all users active. This implies that the migration will take more than one night or weekend to accomplish, thus, it requires a coexistence model. Such a coexistence model enables the overall system to keep accepting users to be added on one platform or the other, to allow passwords to be modified, to accept mail messages to be delivered either to the source platform or to the target platform (depending on whether the user was migrated or not), to handle the situation when the user is being migrated, and so forth.

Figure 2 illustrates the offline model.

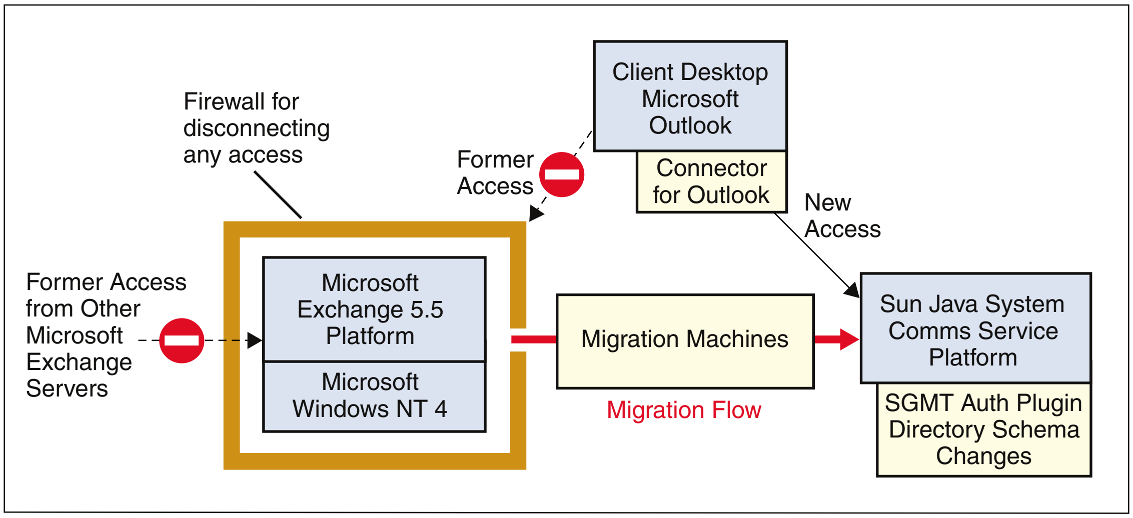
Firewall for disconnecting any access
Client Desktop Microsoft Outlook
Connector for Outlook
Former Access
New Access
Former Access from Other Microsoft Exchange Servers
Microsoft Exchange 5.5 Platform
Microsoft Windows NT 4
Migration Machines
Migration Flow
Sun Java System Comms Service Platform
SGMT Auth Plugin Directory Schema Changes

FIGURE 2 Offline Migration

Figure 2 shows how the source Microsoft Exchange service is put into an offline status. Clients are no longer able to access this service while the migration takes place. When access is restored, it is only to the new Sun Java System Communications Services service.

Figure 3 illustrates the online model.

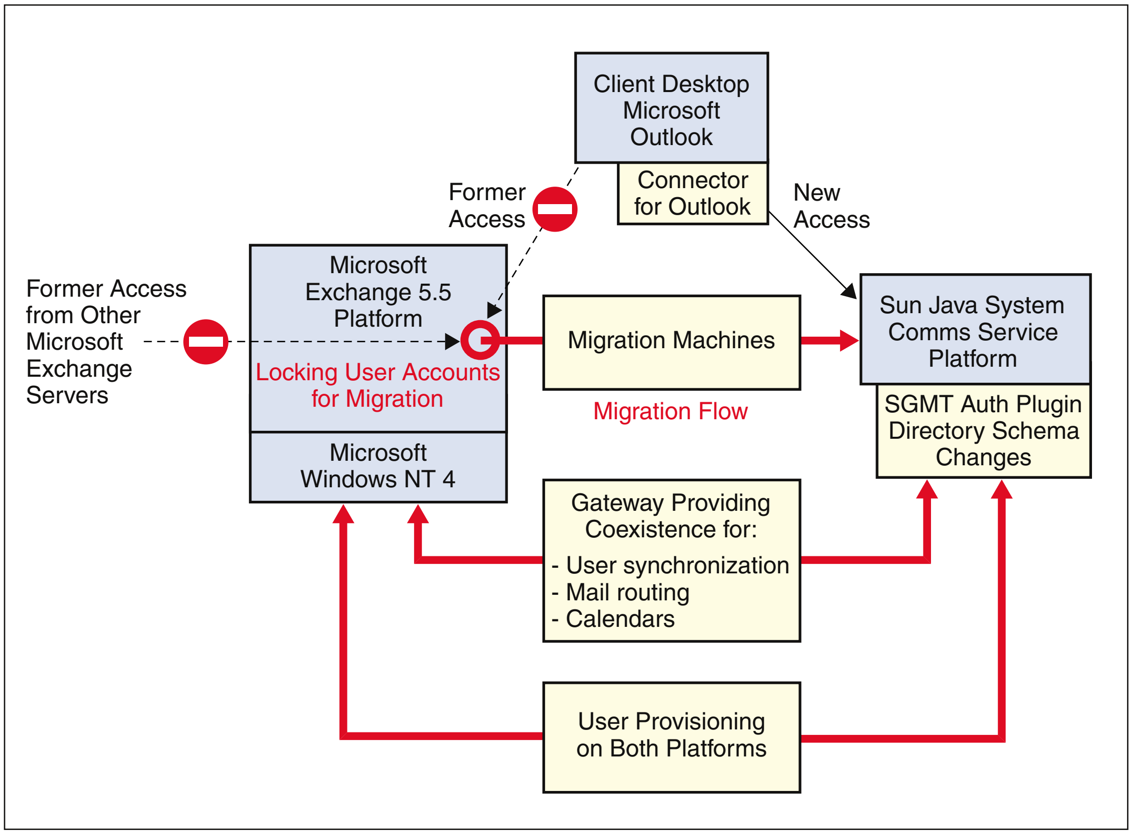
Client Desktop Microsoft Outlook
Connector for Outlook
Former Access
New Access
Former Access from Other Microsoft Exchange Servers
Microsoft Exchange 5.5 Platform
Locking User Accounts for Migration
Microsoft Windows NT 4
Migration Machines
Migration Flow
Sun Java System Comms Service Platform
SGMT Auth Plugin Directory Schema Changes
Gateway Providing Coexistence for:
- User synchronization
- Mail routing
- Calendars
User Provisioning on Both Platforms

FIGURE 3 Online Migration

Figure 3 shows that migration occurs with users active, and in the presence of coexistence between the old and new platforms.

Most customers prefer an online migration model. However, if costs are a concern and if the source systems support a small amount of users with small data sets, you should consider offline migration.

In the online model, the source service is never stopped while end users are being migrated to the new platform, thus no service interruption occurs. For both the Microsoft and Sun Java System platforms to coexist while the migration of the users occurs, a coexistence model is necessary. This coexistence model covers such aspects as user and group directory level synchronization, password synchronization or just re-initialization, and the delivery of mail messages to their correct destination, including routing of the internal Microsoft Exchange 5.5 message flow.

How the SGMT Toolkit Works

When you perform the actual migration, the migration toolkit makes for easy selection of users and dispatching them in the migration technical workflow. The following figure presents a high-level view of how the migration works for the various protocols involved.

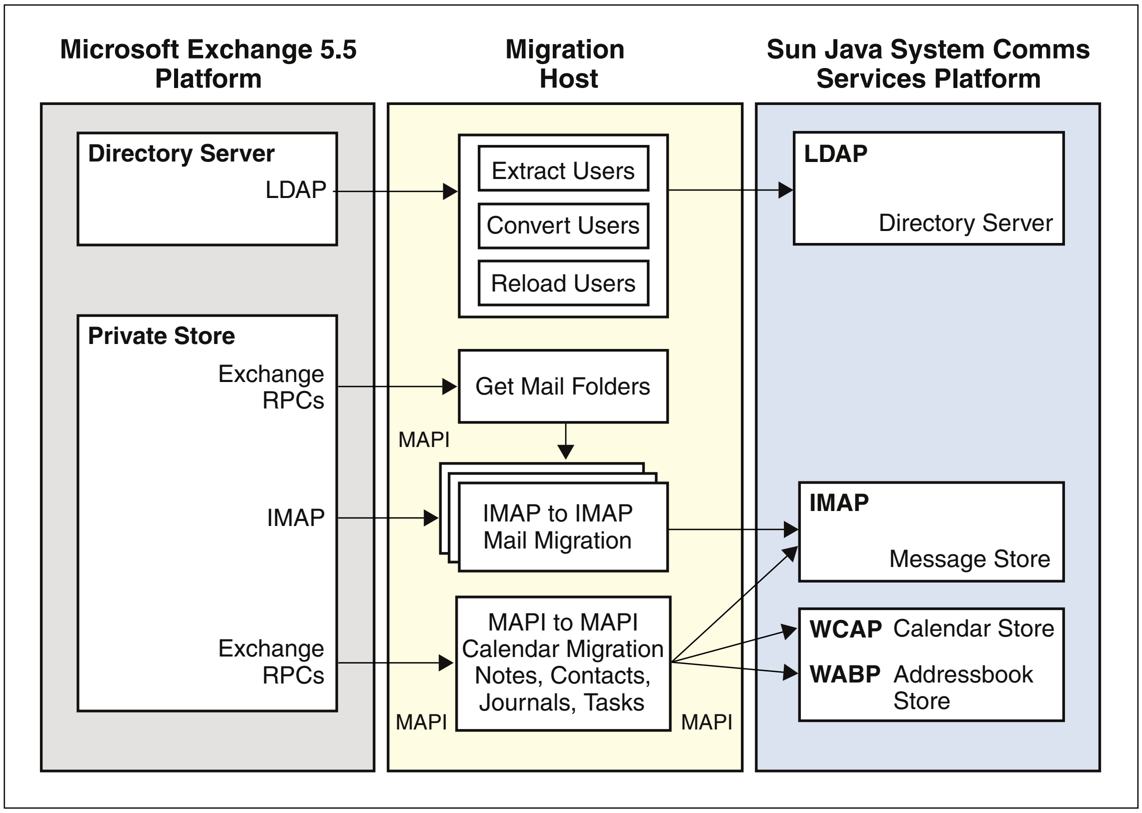

FIGURE 4 Migration Process and Protocols

The toolkit itself performs the following actions:

- Enables you to remap users, mailing lists, and remote users to the Sun Java System Directory Server LDAP.
- Re-initializes the password without the need to know it, “crack” it, substitute it, or ask for the user to change it.
- Sets up coexistence at the user, password, mail, and public folders levels, and also for some aspects of calendar information.
- Enables online migration to take place without loss of mail messages, without the need to stop the full service, and without data storage expansion.

On a per user basis, the toolkit performs the following actions when the actual migration runs:

- Locks the account on source and target platform
- Sets mail coexistence and holds incoming mail
- Scans mail folders, applies remapping rules, and extracts and remaps folder ACLs, including repairing incorrect Microsoft Exchange access control lists
- Migrates mail messages, including repairing incorrectly formatted Microsoft Exchange messages
- Migrates calendar events, tasks, private contacts, notes, and journals
- Unlocks the account on the target platform

More About MAPI to MAPI Migration

Unlike mail, which uses the IMAP protocol, the only way to currently read and migrate calendar, task, contact, notes, and journal data is by using the MAPI standard. The Sun toolkit makes use of two MAPI implementations to accomplish the task of migrating this kind of information.

First, the toolkit uses the MAPI implementation for Microsoft Exchange that transforms any MAPI call into Microsoft Exchange RPC accesses to the Microsoft Exchange server. Second, the toolkit uses a MAPI library to transform the call into the appropriate IMAP, WCAP, WABP, or LDAP call depending on the different Sun Java System server involved.

In principle, to migrate calendar, tasks, contacts, notes, and journal data, a MAPICopy program walks through each object in each folder to:

1. Perform a ’MAPI Copy’ of the object from the interface
2. Read the data from Microsoft Exchange
3. Perform a ’MAPI Paste’ of the object on the interface
4. Copy the data to the Sun Java System servers with the appropriate protocols

The MAPICopy program works without changing anything in the migrated object because the object remains the same in an homogeneous MAPI context.

Figure 5 illustrates how the MAPICopy program takes objects from the source platform and transfers them to the target platform.

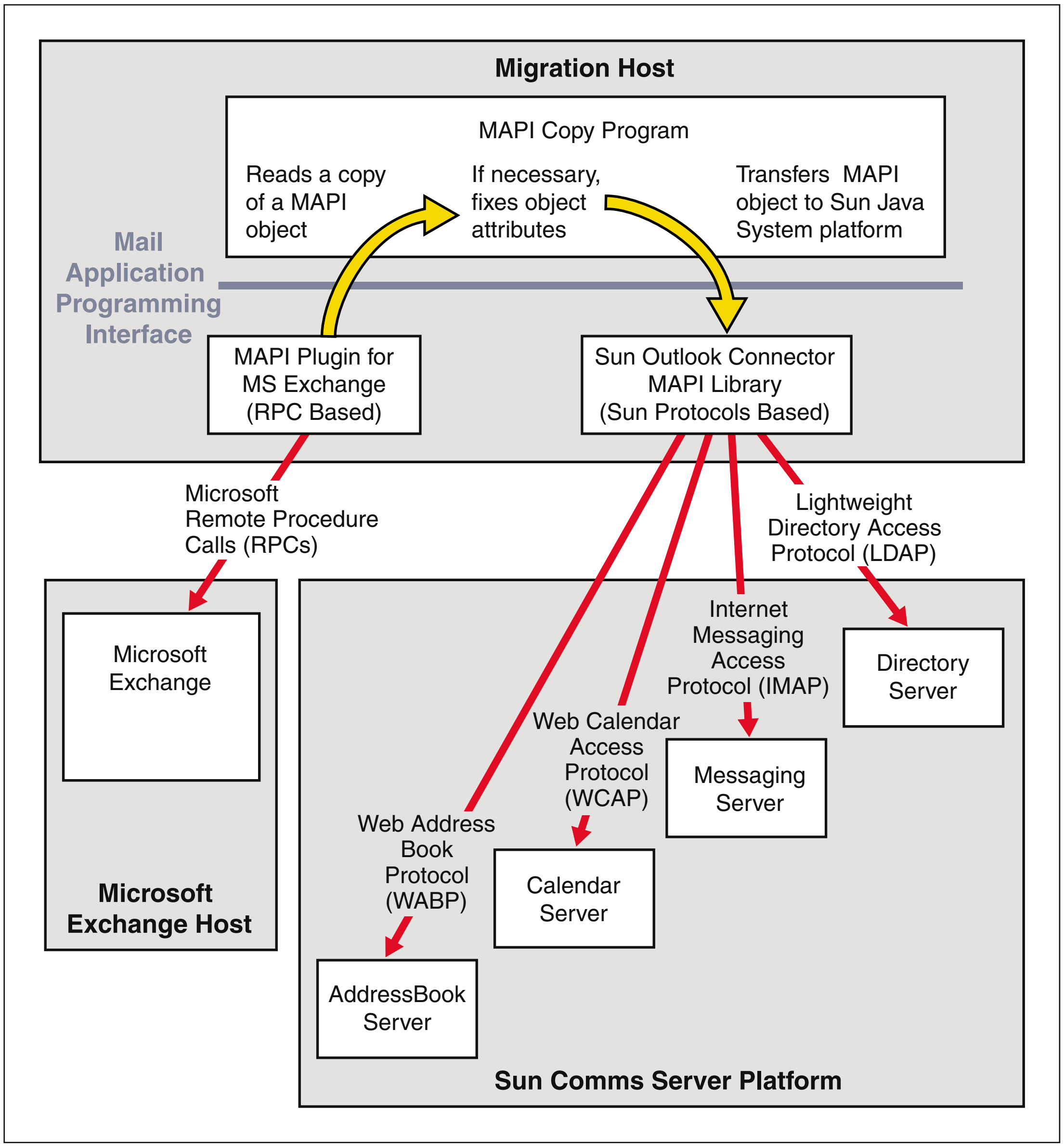
Migration Host
MAPI Copy Program
Reads a copy of a MAPI object
If necessary, fixes object attributes
Transfers MAPI object to Sun Java System platform
Mail Application Programming Interface
MAPI Plugin for MS Exchange (RPC Based)
Sun Outlook Connector MAPI Library (Sun Protocols Based)
Microsoft Remote Procedure Calls (RPCs)
Lightweight Directory Access Protocol (LDAP)
Microsoft Exchange
Microsoft Exchange Host
Internet Messaging Access Protocol (IMAP)
Directory Server
Web Calendar Access Protocol (WCAP)
Messaging Server
Web Address Book Protocol (WABP)
Calendar Server
AddressBook Server
Sun Comms Server Platform

FIGURE 5 MAPICopy Program

Items the SGMT Toolkit Cannot Migrate

Currently, the toolkit cannot migrate the following:

- **Forms.** Contact your Sun Partner for help in this area. Usually, this is a good opportunity to understand why you are using forms, and how they can be streamlined. In addition, you might want to consider using Sun Java System Portal Server and new SeeBeyond offerings in place of forms.
- **Server rules.** Due to folder remappings, it is impossible to correctly recompute server rules in advance of the actions. Normally, this is not a problem, as you can easily recreate the server rules on the target system. This is also a good opportunity to force a review of those rules.

What Factors Impact Migration?

There are no particular restrictions for migration. However, experience has shown that not all formats are appropriate to every situation. There are many aspects of a migration to consider that can influence whether to use an offline or online model, whether to leverage Sun partners, and so on. Consider the following factors when determining your migration:

- **Number of users.** Small companies with fewer than 100 users and simple requirements might want to use an offline migration model.
- **Existing Service Level Agreements.** Companies that have mission critical SLAs can only use an online migration model, ensuring no end-user service interruption.
- **Developing New Service Level Agreements.** Companies might want to take the opportunity to revisit their mission-critical service under an SLA. You might even want to outsource support to a business-to-business messaging hosting platform from the Sun customer network.
- **Identity Management.** Companies that have or want a more mature provisioning model can use this opportunity to revisit their identity workflow. Consider starting a Sun Java System Identity Manager project prior to the migration project. For more information, see: `http://www.sun.com/software/products/communications/partner_library/index.xml`
- **Third-party Applications.** Companies that use third-party or in-house applications on Microsoft Exchange should consider rehosting these applications with Sun Java System Portal Server and Sun Java Integration Suite offerings. You need to review specific applications, such as voice-based systems, unified messaging, integration with fax, and so forth, on a case-by-case basis.
- **Distributed Topology.** Companies with most or all system components and messaging processes distributed across multiple sites will want to consolidate. This process requires pre-project work. In particular, data centers need to consider networking, security, and even disaster recovery.
- **Unsolicited Bulk Email (UBE).** You might want to take this opportunity to include revisiting your anti-virus and anti-spam plan. Sun Java System Messaging Server offers many anti-virus and anti-spam tools, including mailbox filtering, address verification, relay blocking, and integration with various third-party anti-virus and anti-spam products.

- **Archiving.** As part of a migration, companies might want to add some archival features (in addition to your backup plan) to their messaging system. You might also want to introduce compliance and content management systems. Sun currently partners with AXS-One on compliance systems. For more information, see:
`http://solutions.sun.com/`
`partner/pd/PartnerDetail.jsp?vendor_id=1019722`
- **Choosing a new desktop.** Companies concerned with support costs, TCO, and so forth, might want to take this opportunity to consider removing the Microsoft Windows desktop itself, and use an alternative such as Sun Java Desktop System.

There is no single answer to cover all of these situations. As such, Sun and its partners take the time upfront during a two-day workshop to assess the particulars of your organization to arrive at the correct way to handle your migration. See "Workshop Phase (Validate Architecture and Methods)" on page 28 for more information.

Addressing Typical Migration Concerns

Within IT computing, migration projects tend to be the most difficult, as they often encompass heterogeneous deployments. Groupware services are no exception. A few years ago, messaging services were categorized as mission critical, and today this is even more so, as you must account for calendaring and address book services as well.

Sun is perfectly aware of the obstacles presented by migration projects. Sun's migration solution is designed to remove typical migration concerns, as explained in the following sections.

Providing for a Safe Migration

The migration process needs to occur safely and present little if any risk to your data. Sun's migration process provides a proven methodology and toolkit that enable you to prepare everything before running the actual migration. In particular, the toolkit scans all objects in the directory server, scans all mail messages in the concept of a meta store, rebuilds messaging statistics, simulates the new disk and user allocation, and collects data on how long the migration will take-all in a non-intrusive mode. Overall, the migration process includes many best practices that were designed to provide for a successful migration.

Ensuring End User Satisfaction

In the context of a Microsoft Exchange migration, the user experience with Microsoft Outlook must not change. Sun Java System Connector for Outlook works seamlessly with the Outlook client to provide the experience with which end users are familiar. The back-end server migration will be transparent to end users.

Complying with and Enforcing Security Policies

Sun's migration toolkit comes with documentation that can help you understand security on the new target system. This documentation enables project leaders to review the security during the migration and also to understand the access control lists (ACL) remappings for mail folders between non-standard Microsoft Exchange capabilities and the standard IMAP ACL normalized model that is implemented in the Sun solution.

Preventing Transcoding of Data

The migration process prevents transcoding of any data content during the migration. You want this kind of behavior from a migration, in case legal issues arise later. As a company, you might need to be able to present evidence showing that your data was not modified during the migration. The Sun toolkit does not modify your data in any way, except where some Microsoft Exchange messages are not RFC 822 compliant, and thus need to be repaired.

Keeping Migration Cost Low

Sun wants to make it as easy as possible for customers and partners to migrate. Sun also wants to make the service side of the cost as small as possible. Thus, there is no 'per user' cost for its migration toolkit.

Keeping Message Storage Requirements the Same

You do not need to increase your storage requirements as a result of the migration. The Sun Java System Messaging Server Message Store uses a single-copy store methodology that is fully functional with migrated messages. For more information, see Appendix A.

Keeping User Passwords the Same

With the Sun migration tool, there is no need to discover or "hack" user passwords, change them, or require users to change their passwords after the migration. Additionally, the migration tool itself does not need to know the passwords to accomplish its work.

Note – Sun estimates that upwards of 50 percent of a company's internal support calls involve user passwords. Forcing users to change passwords as part of a migration strategy would severely impact a company's help desk. This would also have many other potentially harmful side effects, as passwords can be used in other areas, such as for single sign on access and for batch processing.

Ensuring Coexistence

Coexistence is ensured by synchronizing the Microsoft Exchange and the Sun platforms. This means that the Sun toolkit treats the coexistence issues up to a certain point, in particular with the following components:

- **Directory.** Can occur as one-way or two-way synchronization for:
  - Mailboxes
  - Distribution lists and groups
  - Custom recipients ('LDAP entries' in Microsoft Exchange that point to external mail address that are not part of the Global Address List (GAL))

  Note – You can also perform a bulk synchronization of the directory level. You must perform customizations at this level through remapping rules. In addition, the Sun migration tool offers documented APIs to flexibly remap anything that you like to the target Sun LDAP server.

- **Passwords.** The Sun toolkit does not need to change the user password. The toolkit operates "cleanly" on passwords, with no need for users to substitute a new one.
- **Mail coexistence.** The Messaging Server MTA is able to "hook onto" the SMTP flow and copy/forward model to ensure that the Microsoft Exchange mail messages are routed to the new platform.
- **Calendar free/busy lookups.** Coexistence enables users to see the free and busy status of their colleagues, however, only migrated users can view the status of non-migrated users. Non-migrated users are not able to view the free/busy status of migrated users. Nevertheless, experience has shown that the fact that it works for migrated users actually encourages non-migrated users to get migrated.
- **Public folders.** Microsoft Exchange public folders are synchronized with Sun shared folders to ensure that migrated users within a community are sharing the same data between the two platforms.
- **Calendar synchronization.** Unfortunately, performing a full calendar synchronization is almost unrealistic, as there are too many conflicting requirements. For example, not only must you add or delete calendar entries, but you need to inspect each and every property, and there is no notion of "who wins" the reconciliation.

Making the Migration Simple

Sun's experience with real-world migrations has shown that once you have completed all the necessary upfront planning, analysis, and configuration, a migration does not need to be carried out by the most technical of a company's support personnel. With a minimum of training, your operators should be able to accomplish the actual migration steps. In practice, where you should spend time to smooth the migration is by:

- Preparing email to end users and an intranet page explaining more details on the migration
- Sending notifications to end users when their migration will occur
- Validating the migration, once it is completed, and analyzing post-mortem issues, if any (usually this just involves reassuring users that their data is there)

Keeping Migration Time to a Minimum

In general, the Sun migration tool only needs five to ten minutes to migrate a normal user's actual data. The real migration bottlenecks occur in the following order:

1. Microsoft Exchange is the first bottleneck. In particular, Microsoft Exchange 5.5 on Windows NT 4 has several design issues that slow down the migration.
2. The second bottleneck can be your network.
3. The third bottleneck can be the migration machine if it is under-sized.
4. A real bottleneck is the capability of your help desk and the support team to correctly deploy Sun Java System Connector for Microsoft Outlook on to the user's desktop. This requires:
    - Configuration at the site level for Connector for Microsoft Outlook.
    - Delivery and deployment of the pre-configured Connector for Microsoft Outlook to the user's desktop. Such delivery requires a additional software in place at your organization, such as Microsoft SMS, or tools such as Alterys.
    - Modification of the desktop login sequence to ensure that the Connector for Microsoft Outlook is synchronized for the first time, just after the user migration has occurred on the back end.

Managing Personal Folder (.pst) Files

Personal folder (`.pst`) files may or may not be an issue for your organization. If the goal is to recentralize end user's `.pst` files onto the new target system, it might be reasonable to propose the relocation of `.pst` files on the new platform as part of the migration.

One potential solution is for end users to upload their .pst files (in non-encrypted form) to a shared file system before the migration takes place. Operators performing the migration reload the .pst files in a special place of a slave Microsoft Exchange server and apply the migration tool with a special mode. End users then receive their .pst files in an ‘archive’ folder hierarchy, which is printable and lists folders in a special area of the Messaging Server.

Other alternatives are possible. Consult with your Sun representative for more information.

Organizing the Migration Project

This section provides the migration problem statement and describes how you need a good methodology to approach the problem. It also describes the five phases of migration, from a situation where no Sun products are in place, to a situation where the migration occurred and the old system is decommissioned.

Migration Problem Statement

A messaging migration project involves many activities, spanning management, technical, and methodological issues. Figure 6 illustrates the high-level view and overall direction of your migration project.

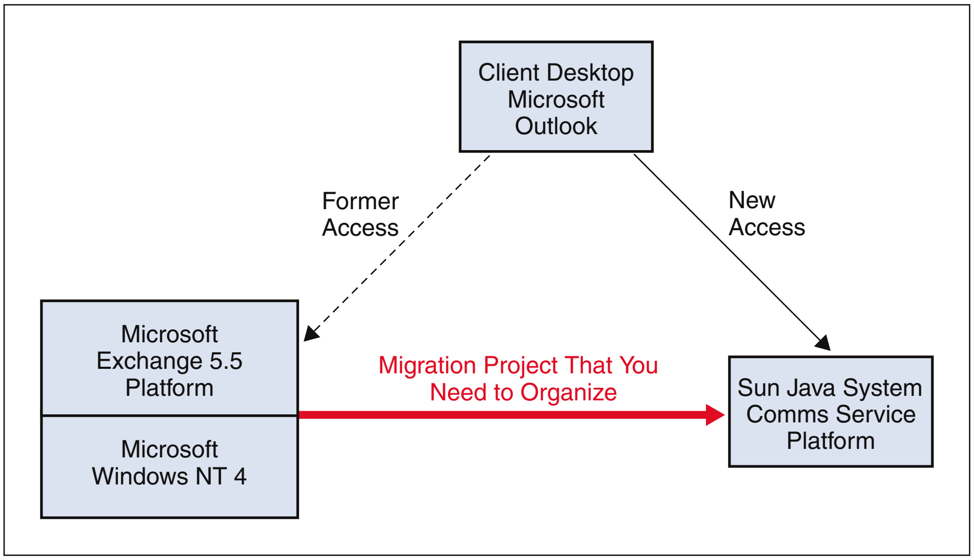

FIGURE 6 Initial State of the Migration Project

Understanding the importance of the migration problem, Sun has developed specific steps (phases) for a migration project to drive a platform from a certain source groupware to Sun Java System Communications Services. This entails understanding:

- Your business, its Service Level Agreements (SLAs), policies (security, archiving, and so on), your organization, and its processes.
- The actual platform running the source groupware system. In particular:
  - Hosting data, general architecture, flows, and so on.
  - User data provisioning, data quality, double provisioning, synchronization, and so on.
  - Disk space usage and other statistics. Specific attention should be paid to the storage expansion rate, which is resolved in the second phase of the project.

The migration project relies on information extraction tools which support some of the steps:

- SGMT provides tools to identify LDAP data issues and to remap user data in a flexible manner. This enables specification and customization of the synchronization between the source and target LDAP services.

  For example, some user login strings can contain characters that are forbidden on the target system. Additionally, because of consolidations in the Directory Information Tree (DIT) into a more global flat tree, duplicate IDs can occur. SGMT detects these problems before and not during the migration such that you can contact users and take corrective actions before migrating.

- SGMT provides tools to compute statistics on users, to simulate their reallocation on the future system, and to evaluate partitions. This information supports ordering or specifying storage requirements as well as refinement of the architecture.
- SGMT then performs the actual migration of the data and provides a migration user status, which allows other systems or clients to take appropriate actions. For instance, the user desktop login scripts can decide to launch Sun Java System Connector for Outlook for each user after—and only after—the user is migrated.

Migration Phases

Figure 7 illustrates the five phases of the migration process and the associated workflows.

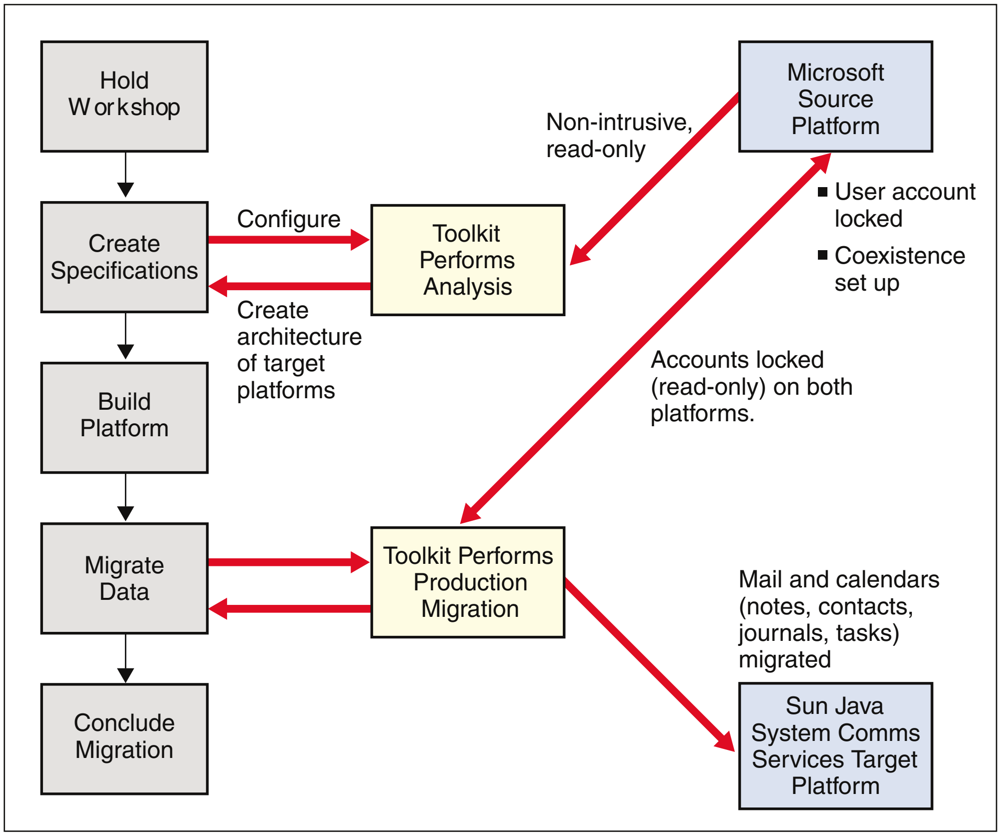
Hold Workshop
Create Specifications
Build Platform
Migrate Data
Conclude Migration
Configure
Create architecture of target platforms
Toolkit Performs Analysis
Non-intrusive, read-only
Microsoft Source Platform
User account locked
Coexistence set up
Accounts locked (read-only) on both platforms.
Toolkit Performs Production Migration
Mail and calendars (notes, contacts, journals, tasks) migrated
Sun Java System Comms Services Target Platform

FIGURE 7 Simplified SGMT Workflow

Workshop Phase (Validate Architecture and Methods)

Prior to this phase, you receive a questionnaire designed to collect information on different technical and non-technical aspects of your organization. After you answer this questionnaire, the workshop phase starts. A workshop is a two-day engagement where two Sun architects instantiate the migration project. The two days are broken into four half-day sessions as follows:

- Session 1: Review of the questionnaire and getting to know the customer.
- Session 2: Review all critical problems and show how Sun resolves them.
- Session 3: Sun articulates the vision and general architecture.
- Session 4: Sun proposes the project format with milestones and dates.

Preferably, the following members of your organization should attend this workshop:

- Project sponsor
- Groupware architect
- One person from the production team (hosting, network, SAN, disaster recovery, data center, geographic dispersion, and so on)
- One person from the security team
- One user representative (for example, from the Help Desk)
- One person from the desktop team

Sun and/or its partners align the two architects, one engagement manager, and one project manager.

After this workshop Sun and/or its partners have all the elements necessary to make a proposal, and a report is written.

Specifications Phase (Design and Plan)

During the second phase, the SGMT tool creates reports on which you base the architecture of the target platforms, as well as migration modalities. As part of this phase, an integration platform is built to support deployment of SGMT for a first-pass analysis and profiling.

Next comes the qualification of your data. Potentially the most demanding task in this part is the analysis of the LDAP data quality. Indeed, all data must comply with the toolkit's requirements (based on RFCs). This task integrates as well the synchronization processes between the source platform and the target Sun platform.

Tip – Do not underestimate the effort required for writing these "mapping functions."

To have an idea of the performance of the source system, statistics are extracted through the IMAP interfaces of the source platform. This process detects as well possible anomalies on the source system, including:

- Mailboxes that do not have uids, uids that are not unique, or invalid characters that have been used
- Folders that were created without a type
- Mail messages that were corrupted on Microsoft Exchange that cannot be read with Outlook
- Folder names that contain a slash (/)
- Calendar items that contain invalid data, duplicate uids, or wrong time zones
- Overly “large” mail messages

These anomalies with source data present key risk factors that must be understood for the migration to be successful.

Once you have collected such data, you can then:

- Specify the allocation models for users on the message stores
- Specify and size partitions
- Simulate disk space allocation before the actual migration, taking into account or not the relinker functionality in the Message Store (to avoid migration storage expansion)

Once all these tasks are completed, you are then able to:

- Customize the toolkit to your particular project’s needs
- Test the customized toolkit
- Drill down into user profile data
- Drill down into user messaging data

Finishing the specification of the overall platform involves performing LDAP provisioning, user allocations, disk space requirements, and so on.

Build Phase (Configure)

During the third phase, you reconfigure SGMT again, this time to perform the migration to the new platform instead of the analysis, which is a read-only operation on the source Microsoft Exchange platform. In this way, the new target platform is built. At this stage the most critical activity is to enable the synchronization of the source LDAP platform with the target LDAP platform. Once you validate this step, users and mailing lists are synchronized between source and target. On the target platform, “shadow” users are ready with the correct attributes for their respective real users’ migration.

During the third phase, you construct components that drive some of the acceptance testing. For example, you need to:

- Set up a test LDAP server, on which you:
  - Add the SGMT schema, password plugin, indexes, and so forth

  - Pass the dirmapping on the current source LDAP server
  - Find and resolve LDAP errors
  - Implement mapping functions and test
- Build a meta store of the source system, and process archival data that exists perhaps for legal or business reasons but that can be reprocessed afterwards, with:
  - A 'per message' md5, size, IMAP uid of mail messages for all users and folders
  - Reporting that enables you to understand users mail usage profiles
  - User allocation preparation and simulation until satisfying rule is found

Migration Phase (Deploy)

The fourth phase enables the actual migration of user profiles and data. In this phase, the following actions occur:

- You have defined how quickly you want your migration to take place.
- Your help desk has been trained and is ready to start answering user questions.
- You have deployed Connector for Microsoft Outlook, and it is ready to launch when the right attribute of the user in the LDAP is set to 'migrated.'
- The migration itself can take place.
- For each migration pass, the following takes place:
  - Select accounts by means of SGMT.
  - Start the migration Dispatcher:
    - The Dispatcher "locks" all selected accounts on both platforms.
    - Mail messages, calendars, tasks and contacts are migrated.
    - Accounts are "unlocked."

Epilog Phase (Close Migration)

The fifth and final phase involves resolving any problems after the migration has been completed. SGMT is still necessary until the source platform is fully decommissioned, for synchronization purposes. However, as soon as the source platform is completely removed from the architecture, SGMT can be uninstalled, and the migration hardware can be re-purposed. You might need to attend to other, overlooked tasks as well, such as removing DNS entries, decommissioning machines, and so forth.

For More Information

For more information on migrating your groupware service to the Sun Java System Communications Services platform, see the following:

```
http://www.sun.com/
software/whitepapers/wp-exchange/wp-exchanging_exchange.pdf
```

```
http://www.sun.com/service/emcs/messagingmigration.html
```

APPENDIX **A**

Appendix

This appendix describes additional information on migrating your messaging platform to Sun Java System Communications Services.

About Message Storage Concepts

When performing a messaging migration, no migration tool has access to the low-level interfaces of the stored mail messages. This becomes problematic when dealing with a message that has multiple recipients and thus might be stored multiple times in the mail storage.

If the messaging system supports it, you would want to store just a single copy of the message, and then through pointers, enable the multiple recipients to reference that single copy, thus reducing overall storage need. That is, you don't need to store a duplicate copy of the message for each recipient. Both Microsoft Exchange and Sun Java System Messaging Server support this single-copy storage scheme.

Statistics show that a typical multiplication factor for a message store in a corporate environment is between five and ten. Obviously, you do not want to increase your message store during a migration from 500 GB to 5 TB of disk space because of this issue.

Note that this issue is faced by all migrations, and does not uniquely apply to Sun. What Sun states is that this is a "non-problem" when migrating to the Sun Java System Communications Services platform. Though one cannot compare directly the message store models of Microsoft Exchange and Sun Java System Communications Services, you only need an equivalently sized storage for the migration.

Figure 1 illustrates the idea of the single-copy message store.

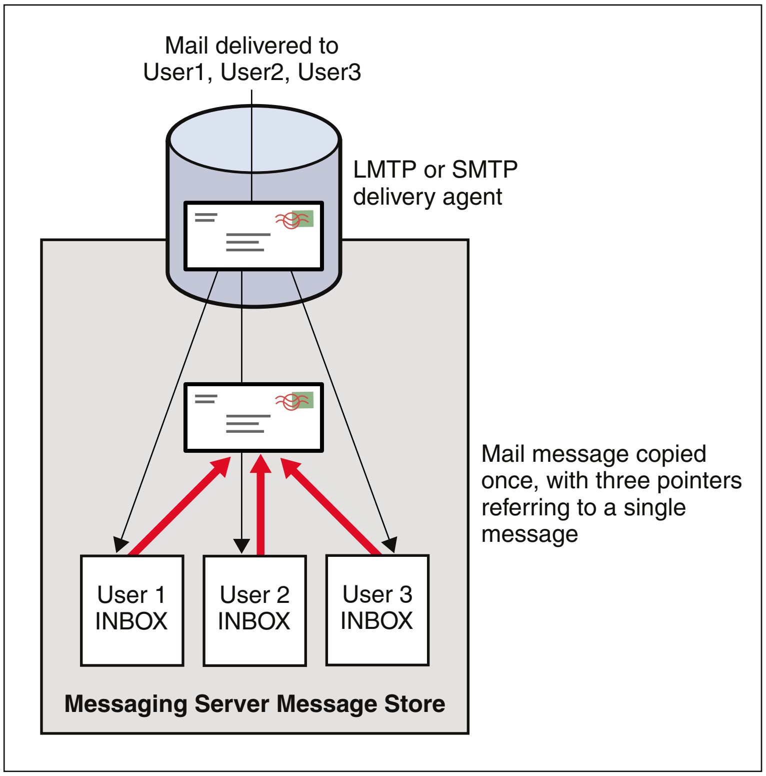

FIGURE 1 Single Copy Message Store

The figure shows that only a single copy of any message is saved in the store as a single file. All other instances of that message (for example, when a message is sent to multiple mailboxes) are stored as links to that copy.

The problem that a migration is faced with in making sure that storage expansion does not occur as it copies these messages to the target platform is illustrated in Figure 2.

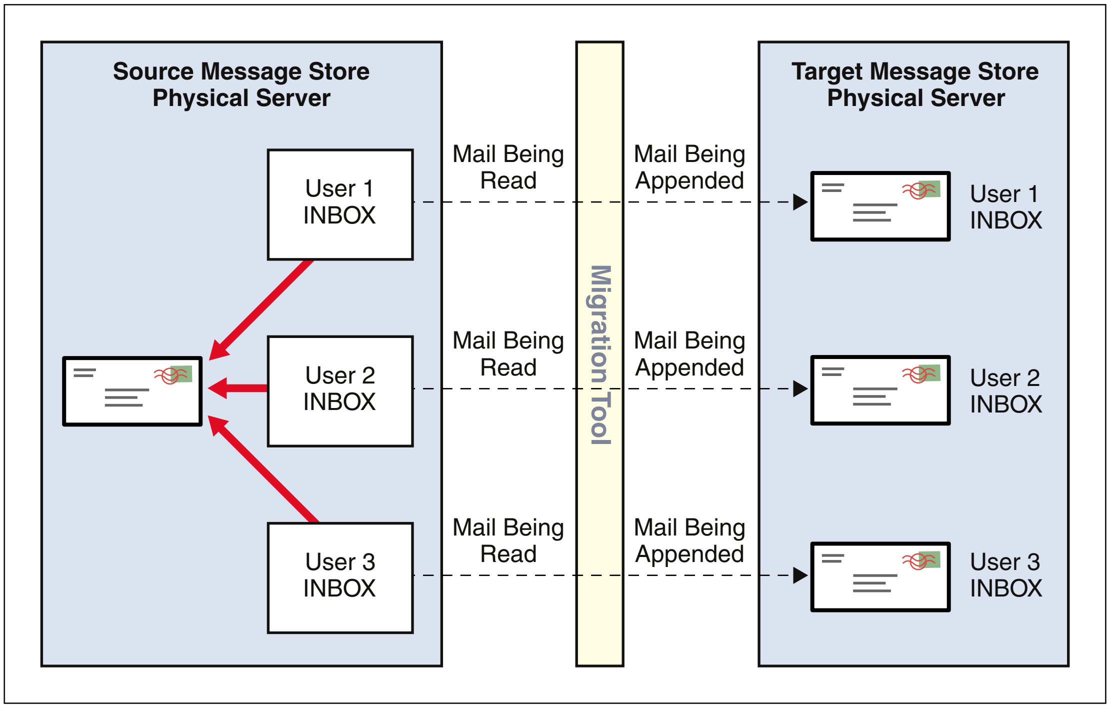

FIGURE 2 Migrating Messages

To prevent storage expansion from happening, Messaging Server contains a relinker feature to remove excess message copies, and instead, use hard links to a single copy of the message. Figure 3 illustrates how the relinker works.

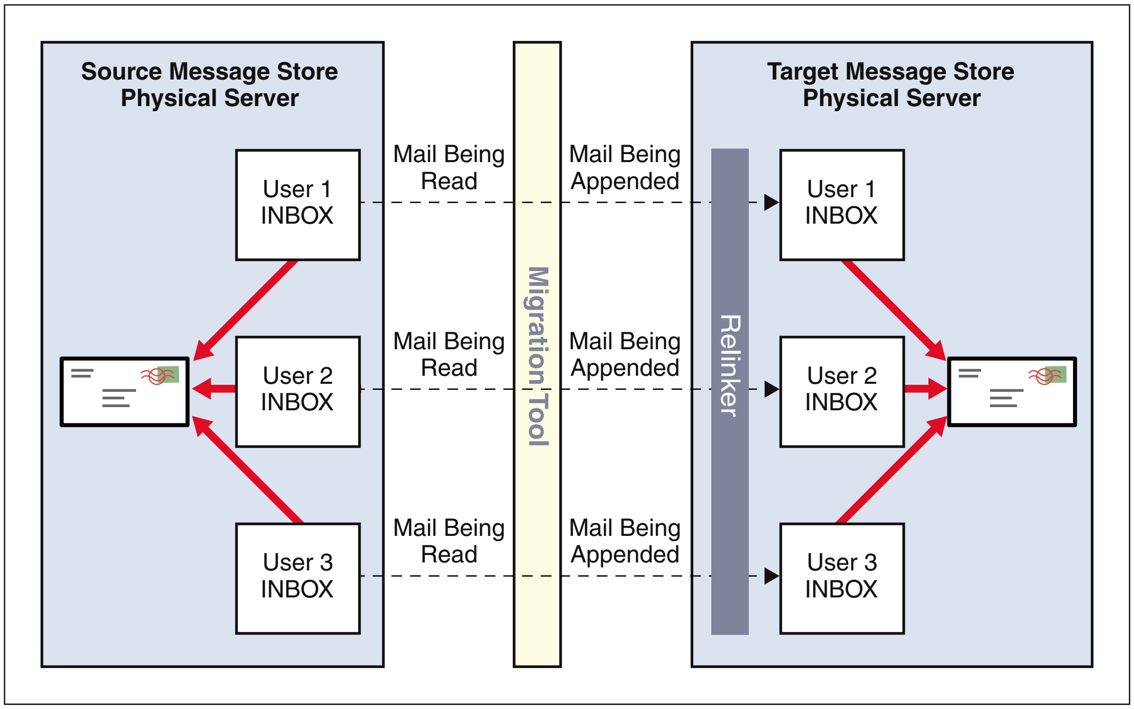

FIGURE 3 Migrating Messages with the Relinker

You can enable this feature in real-time or run it as a separate command-line utility. The relinker enables you to identify those messages that are identical and to relink them in the store. In this way, you do not need to increase your storage requirements for your messaging service.

For more information on this topic, and on the Messaging Server relinker, see "Relinker Theory of Operations" in *Sun Java System Messaging Server 6 2005Q4 Administration Guide*.