Desktop Architecture Selection Guide

Howard Carlton—Sun Professional Services

Sun BluePrints™ OnLine—October 2003

http://www.sun.com/blueprints

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95045 U.S.A.
(650) 960-1300

Part No. 817-3993-10
Revision 06, 10/10/03
Edition: October 2003

Please
Recycle

Desktop Architecture Selection Guide

Over the past ten years, most companies have opted for a distributed desktop model engineered so that each desktop has the same hardware and software configuration and provides the same services (for example, office and business applications), irrespective of the line of business for which it is intended. This *one-size-fits-all*, or thick client, strategy is expensive and complex to operate. Administrators have to support whole populations of PCs, replicate frequent software upgrades, fix PCs at the user's desk, and cope with issues created by security flaws and the need for distributed backups. Add to this the hidden costs of user *self-support* and lost productivity through failed systems, and the total cost of ownership (TCO) becomes considerable.

Recently, users and managers who are required to fund desktop infrastructures have become increasingly conscious that placing the processing power and complexity of a 1980's mainframe on every user's desk is not necessarily the panacea that it might have seemed a few years ago. Some of the negative factors influencing this thinking are:

- Cost of ownership

  The rate of change of the underlying hardware and the frequent rerelease of costly software products are forcing IT departments and users into expensive upgrade cycles.

- Manageability

  Each thick client solution contains a complex set of hardware and a fully-fledged operating system, which in turn support an application stack. When failures occur or software products need to be updated, there is a significant requirement for relatively expensive resources (that is, IT support people) to diagnose problems, fix the problems, and rebuild the systems.

- Security

  It would appear that the most popular systems are those most frequently subjected to malicious attacks. The daily release of security bulletins from the authors of these systems reveal the high level of vulnerability in some of their products and the near impossibility of creating a secure environment based on such products.

- Reliability

  Complexity and component count are the primary drivers of failure rates. By placing large numbers of sophisticated systems on desktops, rather than smaller numbers of more manageable systems in the data center, companies are ensuring that failures will be more frequent and harder to resolve.

Many organizations are actively pursuing alternative solutions that can remove or ameliorate some of these obstacles to user satisfaction and corporate progress. In many instances, users are evaluating alternative application-access devices that enable them to move toward a more controlled and server-centric service delivery environment. Candidate replacements include the large number of browser-based and Java™ technology-enabled devices such as 3G phones, PDAs, and thin client devices. As the availability of web services increases and as organizations move to web-based delivery of their in-house applications, such devices will have adequate computing and presentation capacity to meet the needs of the vast majority of users.

At a time when IT budgets are under close scrutiny, it is prudent for IT and IS managers, planners, and strategists to evaluate the potential of reducing cost and complexity by deploying the server-centric paradigm and to be prepared to discard some of their legacy environments. This article provides information and pointers to help designers to shape less expensive and more reliable desktop architectures.

Current Approaches to Desktop Architectures

Recent approaches to desktop architecture address the problems of a thick client architecture by using a centralized model in which applications are delivered as services to users on any device. This gives the IT organization the flexibility and control to deliver only those services that are necessary for the individual user. Thus, overall TCO is reduced and manageability improved.

Built around the simplicity of thin client devices such as the Sun Ray™ ultra-thin client, a centralized desktop architecture provides access to applications running on Microsoft Windows, the Solaris™ Operating System (Solaris OS), and other platforms from a single desktop. Users can access the applications they need, and administration of the system is simplified. The desktop runs no software or

operating system, so it never needs upgrading. Users can also choose the ease-of-use and low cost of alternative, open source productivity software such as the StarOffice™ Office Suite and Mozilla™. In the data center, a consolidated server solution runs the operating systems so that updates happen at one place and are immediately available to all desktop users.

Approach to Implementation

It is not sufficient from an implementation perspective to simply select hardware and software alternatives. Organizations need to evaluate, design, and plan the whole life-cycle of their desktop environment, and project teams need to address the three vital life-cycle components: their people, their processes, and their technology.

Organizations need to start with a comparison between their business needs and their current desktop environment to identify opportunities for improvement and to recommend suitable alternative solutions. From this evaluation process, a clear, phased approach can be identified and agreed upon with key decision makers. Sun and its partner organizations can provide specialist consultants to assist with the processes of architecting, proving, and implementing an appropriate solution to ensure that the TCO for the proposed desktop environment represents an appropriate and necessary solution for the organization's business. Sun and its partners can also provide managed services consultants to design and, in some cases, operate the solution during an interim or hand-over period.

Architecture Selection

Selecting an appropriate desktop architecture is a complex and multifaceted task. The needs and attributes of every *real world* customer environment have some unique properties, as well as many factors in common with other users. In addition, very few sites are *green field*. There is usually an existing infrastructure that represents significant investments in hardware, software, and technical skills. Changes to such an environment are generally evolutionary rather than revolutionary. In times of slower economic growth, CFOs and other executives are reluctant to sanction expenditures that do not show an early return—typically in the form of a reduction in outgoing costs.

The purpose of this section is to provide an overview of some of the main factors and guiding principles that architects and CIOs tend to apply when considering changes in desktop architectures and also to examine some factors that relate to specific user types and industries that complement the overall guidelines. Finally, a summary of the main architecture *patterns* and their applicability to a selection of environments is provided.

Decision Areas

There are three major areas of choice confronting the designer of a desktop architecture:

- Client-centric versus server-centric computing
- Proprietary versus open source operating platforms
- Client device type(s)

The factors that need to be considered when making these choices are also the primary drivers that have to be understood and optimized by IS and IT management:

- Total cost of ownership (TCO)
- Manageability
- Security
- Usability and performance

These factors are not necessarily independent. For example, manageability and security could have a significant impact on TCO, as discussed on page 6 and page 7 respectively.

Total Cost of Ownership

TCO is driven by a complex mixture of factors. The key issues to be considered by decision makers are:

- Hardware costs

  The cost of acquisition is usually the *headline* number that most people consider when thinking about hardware costs, but hardware lifetime (which might be dependent on operating platform choice), hardware maintenance costs, installation costs, and move or change costs must also be assessed. Although thin-client solutions might appear attractive when rated against most of these items, they typically require a significant investment in server infrastructure, which must be factored into the overall equation.

- Software license costs

  Software license costs can be more complex than they might appear. The list price of many proprietary products appear much higher than their open-source equivalents, but many organizations are tied into medium-term software contracts that could be expensive to cancel. In many cases, these contracts involve a bundle of products, so the substitution of a single open source product for any one of the bundled applications might not be as cost-effective as a straight comparison of one-off license costs might suggest.

Organizations need to be sensitive to the timing of contract renewal to determine at what point a change of software strategy might become viable. Other factors that need to be considered when examining switching costs are the people and process changes that might be required. Generally, some user retraining and support will be required in the transition period, and in some cases, there will also be a need to do document, data, or template conversions.

- Management, support, and security costs

  The people costs associated with the management and support of desktop users are usually the largest single line item of directly attributable costs. These topics are discussed in detail on page 6 and page 7.

- Data storage costs

  Decisions on data management solutions are often taken independently of desktop design tasks, but they have a significant impact on the overall costs of user support. Consolidated solutions such as network attached storage or storage area networks would appear to be a good fit with server-centric (that is, thin-client) desktop architectures and can significantly reduce storage costs by:

  - Increasing the efficiency of storage utilization, as compared with directly attached storage solutions, eliminating wasted space and also avoiding localized shortages
  - Reducing the time and costs required to complete backups
  - Ensuring that data is stored centrally on resilient hardware and is backed up regularly (the cost of data loss and time to recover lost data can be significantly reduced)

- Power and cooling costs

  These cost might not seem to be a significant factor at first glance; however, studies at Sun have shown that for 25,000 users, a move from a thick-client environment, in which every user has a fully functional workstation, to a thin-client architecture will save around six million dollars a year in power and cooling costs.

- End user costs

  Generally, CFOs are reluctant and unwilling to include in financial justifications the benefits of process and technology changes for end users. However, significant *soft* benefits can accrue from a move to server-centric systems:

  - Users typically experience greater uptime as a result of greater hardware reliability (that is, fewer components to fail), simplified software configuration (for example, less chance for a user to misconfigure the system), and faster boot time (for example, *instant on* versus *always on*).

- Users do not need to give up their machines for significant periods of time for IT personnel to carry out software upgrades.
- Users are generally unable to introduce their own software and/or make unauthorized copies of data or applications in an environment in which they have limited access to storage media and peripheral devices. This might seem to be a trivial point, but a significant impact on overall effectiveness and TCO can be caused by a minority of users who abuse the availability of corporate systems and resources.
- Users of the Sun Ray ultra-thin client technology also have the ability to move their sessions (and hence their work) from location to location merely by inserting their Java Card™ into the nearest device. This facility can operate across continents and represents a great convenience for mobile users who can thus avoid carrying heavy laptops.

Manageability

Manageability is a key driver of TCO in most organizations and a significant topic in its own right. Significant components of cost relating to user and device management are:

- Number and variety of devices and platforms that require support

  Operating system proliferation at both the client and server levels tends to drive up costs exponentially due to the diversity of skills and number of support personnel required.

- Frequency of hardware, operating system, and application changes

  This is a significant issue in most organizations operating a heterogeneous thick-client architecture because the time taken to qualify a new hardware or software version can be significant. In addition, both users and IT personnel must make significant investments in time and effort to carry out upgrades or to apply patches for security updates at the rate of one or a few machines at a time.

- Help desk and desk-side support

  This support tends to involve greater time and effort for thick-client architectures. Users whose systems are not *locked down* tend to create problems arising from operating system misconfiguration. If problem resolution requires a desk-side visit, clearly the cost of resolution rises significantly.

To counter these and other cost drivers associated with system and network management, organizations have adopted a number of improvement strategies:

- Implementation of management tools

  Implementation and management tools enable help desk staff to remotely diagnose and fix user problems.

- Automated software distribution

  A number of products and/or system integration processes are available to enable user systems to be updated automatically, usually on a *push* basis.

- Migration to thin-client solutions

  A thin-client architecture can alleviate most of the issues outlined previously. In many organizations, the first move into a thin-client architecture has been to *repurpose* existing hardware platforms by the addition of appropriate software products (see "Services and Servers" on page 21).

- Outsourcing desktop support to third parties

  This is often an attractive solution for business managers because it moves the problem of controlling costs and managing IT staff (which are potentially scarce and/or expensive resources) to an outside agency, while achieving greater predictability of IT budgets. Potential drawbacks of such contracts, however, are that it might be difficult for the client organization to migrate to a new and potentially more cost-effective architecture. The IT outsource supplier might be reluctant to reduce its revenue stream and pass on cost savings to the client because this would conflict with its own business model and objectives. In addition, the support level might drop, reducing productivity.

Security

A wide-ranging discussion on security topics is outside the scope of this article. There are, however, a number of security issues that are particularly relevant to desktop environments:

- Virus infection

  Virus infections potentially have a significant impact on TCO, as well as user confidence.

- Federated identity

  Federated identity might be required for support of a mobile user community.

- User identification

  User identification includes additional authentication mechanisms, such as Java Card technology and challenge and response systems.

- Single sign-on

  Single sign-on is the capability to *pass through* a user's identity to a variety of applications and services.

- Data security

  Data security is the need to protect data on behalf of users and to recover from storage device failure.

These issues are discussed in more detail in “Nonfunctional Requirements” on page 30. For more general information on security, go to:

`http://www.sun.com/solutions/blueprints/`

Usability and Performance

The selection of appropriate client devices is driven in part by overall organizational needs and standards, with a move toward thin clients being fairly generic at present. A recent study by the Giga Group claims that server-based computing has been growing at a rate of 35 to 50 percent per annum and will continue to grow (albeit at a slower rate) for at least the next two years. For more details on the study, refer to:

`http://www.citrix.com/site/NE/news/news.asp?newsID=8121`

Nonetheless, the world of client devices is not a *one-size-fits-all* environment. Specifically, the following user types do not fit easily into the thin-client paradigm:

- Users with heavy graphics requirements

  These user might include those that need powerful visualization software in mechanical engineering and design departments and those that need streaming video such as training departments and media companies. There is a potential trade-off in these situations because many thin-client devices have good two-dimensional and reasonable three-dimensional performance, but they might need access to significant processing power to function effectively as design workstations. On the other hand, a user of a server-centric computing solution potentially has access to greater overall compute resource than the users of single CPU client systems, especially at times when other users are less active. Grid computing solutions offer a possible solution in many of these situations.

- Mobile users

  Most organizations have a number of mobile users, such as sales and service people, who might not have access to the corporate network for much of the time and who need an easily portable access solution.

Industry Preferences

A final set of factors that might come into play when considering desktop architecture options are the preferences and defacto standards that might apply to the particular industry sector or public sector to which any given organization belongs.

Many public bodies are more likely to consider open source solutions. This trend is probably a combination of cost sensitivity (that is, public bodies have a duty of care when disbursing tax dollars on products that do not provide an immediate public benefit) and a philosophical disinclination to favor *big business* over the efforts of open source contributors who are not expecting financial gain from their efforts. This tendency is very strong in Germany and other northern European economies (for example, Sweden, Denmark, and to some extent, the UK), but it shows signs of spreading to APAC and the Americas.

Other cost-sensitive sectors that might be inclined to look at reducing TCO (whether by migrating to open source products or by other means) are driven less by altruistic motives and more by restrictions on capital spending. Generally, this group includes the manufacturing sector (particularly mechanical engineering environments), primary production (for example, utilities and forestry), and the construction industry. Although the financial services industry is passing through a phase of consolidation and reduced capital spending, it has always had a high need for security, accuracy, and timeliness of data and is more inclined overall to spend on solutions that meet these objectives. The communications and media industries not unnaturally tend to be early adopters of new technologies and are likely to be the widest users of mobile devices and innovative access solutions for the next few years.

Architectural Options

The following table summarizes some of the observations made in this section and gives likely solutions that meet the needs of users in a variety of organizations.

TABLE 1 Architectural Options Per Usage Category

Usage Category	Architecture Options
Large, public sector, office-based users (for example, government, education, and defense)	• Open source software • Thick client with *locked down* configurations, with life extended by deploying Linux • Some thin-client *islands* for training, education, and public access (for example, libraries and Internet cafes) • Cost-sensitive enterprises (for example, manufacturing, utilities, and retail) • Potential large-scale thin-client users • Specialist devices for some heavy three-dimensional graphics users • Consolidation of servers and storage for TCO reduction
Financial services, telcos, media organizations, and e-business	• Mixture of thick and thin clients because some users have streaming video requirements • Consolidation of servers and storage for TCO reduction • Significant mobile device support requirement with security, Virtual Private Network (VPN), and Java Card technology • Federated identity services
Collaborative workers in any of the mentioned sectors (for example, call centers, help desks, banks, and manufacturing facilities)	• Good candidates for thin clients with session mobility • Possible Voice Over IP integration with thin-client devices • Might need Computer Telephony integration and Customer Relationship Management integration

TABLE 1 Architectural Options Per Usage Category *(Continued)*

Usage Category	Architecture Options
Mobile workers who are infrequently in the office and users who are mobile within a site (for example, maintenance staff and construction workers)	• Mobile device support • Portal access to applications and data • Security solutions (for example, VPN and Java Card technology)
Small offices and home offices	• Open source software • Laptop and mobile devices • WAN-based thin-client devices • Portal access to applications and data • Security solutions (for example, VPN and Java Card technology)

Architecture Components

This section explores the major components of desktop architectures, their applicability to varying user contexts, and some elements of component design and scaling. The four major components of the desktop environment are client devices, application integration solutions, services and servers, and software. The following diagram shows a typical heterogeneous customer environment containing many of these elements, which are in the following sections.

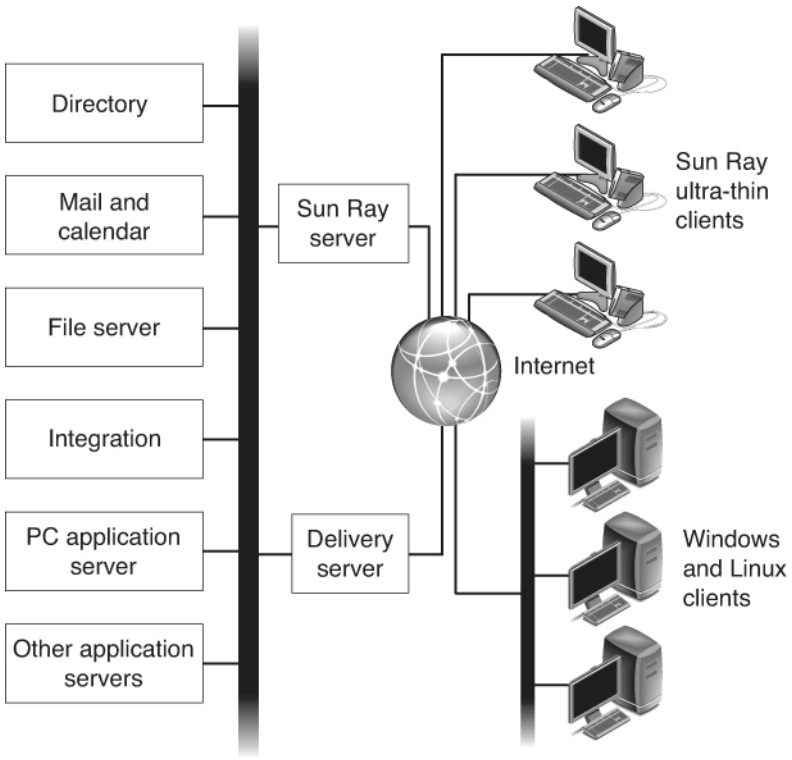

FIGURE 1 Desktop Environment Components

Client Devices

The range of potential client devices has grown in recent years, and desktop or mobile users have a wide choice of interfaces to their business and personal productivity applications. There is also an emerging trend for organizations to adopt a thin-client strategy to manage and reduce their TCO. This section examines client device options (both thick and thin) and gives guidelines on how to select an appropriate device for specific user groups.

Sun Ray Ultra-Thin Clients

The Sun Ray ultra-thin client consists primarily of the hardware and firmware required to present the output associated with a user's desktop session onto an integrated or detached display device (either CRT or LCD) and to accept user input in the form of keystrokes and mouse movements. The actual processing for the

user's session takes place on an associated server that is running the Sun Ray ultra-thin client server software and provides the CPU, memory, and storage resources needed by the user.

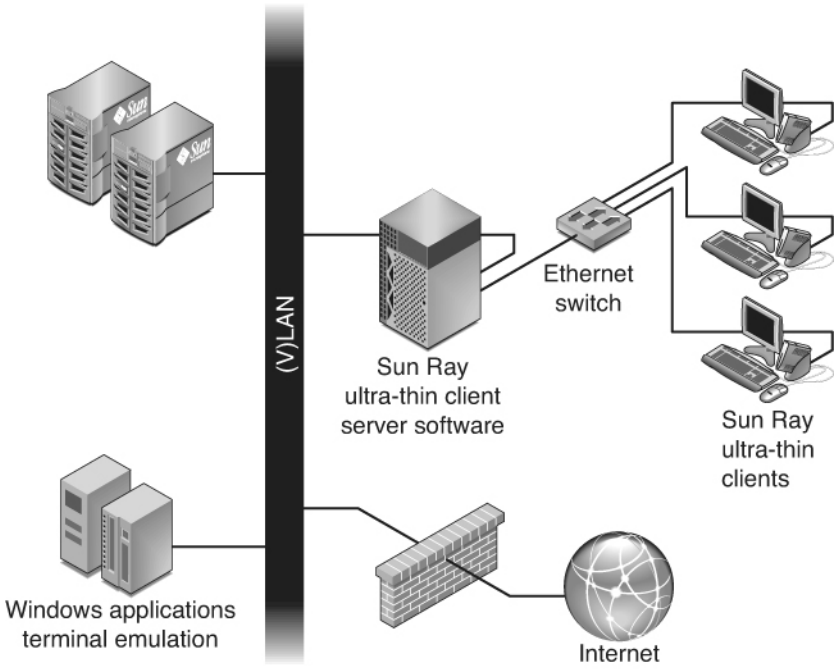

FIGURE 2 Sun Ray Ultra-Thin Client Architecture

The Sun Ray ultra-thin client moves desktop processing back into the data center or local server cabinet, providing for scalability, resilience, and centralized storage and backup. The other significant feature of the Sun Ray ultra-thin client is the ability to relocate sessions between devices. This feature is dependent on the association between a user and a Java Card technology, which carries a unique token used by the server software to identify the session. At present, this session mobility feature is limited to devices attached to servers in the same physical location, but in the future, it will extend to devices located anywhere on an enterprise's WAN.

The Sun Ray ultra-thin client requires bandwidth of around 1 megabit per second between itself and the server. This requirement is likely to be reduced with future versions of the Sun Ray ultra-thin client software and firmware.

Benefits of the Sun Ray ultra-thin client include:

- Long lifetime due to the simplicity of construction

  With fewer components, the Sun Ray 1 ultra-thin clients have fewer failures and are supplied with a five-year warranty.

- Low power consumption and reduced cooling requirements
- Reduced help desk intervention to solve user misconfiguration, operating system problems, and disk crashes
- Ease of installation and reconfiguration

  The Sun Ray ultra-thin client receives its identity (that is, its IP address) from a server after the client is powered on. Therefore, the client can be installed or relocated by unskilled staff or by end users.
- Session mobility

  This is a convenient feature for collaborative workers, mobile users, and users who want to deliver presentations.

Drawbacks of the Sun Ray ultra-thin client include:

- Continuity of service

  The client is dependent on server and network resilience.
- Availability or suitability

  Currently, the Sun Ray ultra-thin client is not available or suitable for small offices (fewer than five users) or home offices until bandwidth requirements have been reduced. This situation is scheduled to change with the next release of the server software, sometime in the middle of 2004.

Windows-Based Terminal Clients

A solution that has been deployed by many organizations to deliver applications to users is to install a simplified (that is, thin-client) device known generically as a Windows-based terminal. The Windows-based terminal solution seems to provide many of the benefits of the Sun Ray ultra-thin client solution without needing to install a server layer (other than Windows application servers). It usually contains a limited software set (operating system, Citrix or Remote Display Protocol [RDP] client, and browser) in firmware, plus local memory and processor, but no local storage facility. Therefore, the Windows-based terminal solution is potentially more reliable than conventional PCs.

Benefits of a Windows-based terminal solution include:

- Reduced complexity and longer working life than thick client systems
- Easy installation and configuration
- Reduced help desk intervention to solve user misconfiguration, operating system problems, and disk crashes

Drawbacks of Windows-based terminals, as compared to the Sun Ray ultra-thin client, include:

- Client device is somewhat more complex

  The lifetime might be shorter than the Sun Ray ultra-thin client.

- Vulnerability to theft

  A Windows-based terminal client might contain *desirable* components, such as flash and conventional memory, which might make the unit more vulnerable to theft.

- Limited capability for session mobility
- No native access to Solaris OS and/or Linux-based applications

Windows Desktop Clients

Traditional desktop systems based on the Intel x86 architecture and the Microsoft operating software have been the clients of choice in many organizations for a number of years. They offer independent processing power to the user and the capability to install a wide range of personal productivity and line-of-business applications that meet the individual needs of the user. A typical enterprise today has more than 1,000 different PC-based applications running on its desktop client systems.

One of the primary drawbacks of these systems is that the IT organization is expected to refresh the hardware and operating software on a frequent basis. The terms of Microsoft's Software Assurance program and the ability of hardware manufacturers to frequently upgrade equipment specifications mean that many users are compelled to refresh their software, at most, every three or four years. For each such refresh, the IT organization (whether in-house or outsourced) will spend many months requalifying the applications that run on the client estate on the updated hardware and/or software platform.

Linux Clients

As the open source movement gains support in the computer industry, it has become increasingly evident that there is a potential market, particularly among cost-sensitive organizations, for Linux desktop systems. Some popular Linux distributions now contain a number of packages that supply the needs of the desktop user. For example, Sun has recently announced a Java Desktop System that includes StarOffice, GNOME, and other open source software packages.

It is now feasible to replicate many or most of the generic facilities provided by traditional desktop clients on a Linux system. A potential desktop client offering should contain the components described in the following table.

TABLE 2 Sun Java Desktop System Components

Component	Product
Operating system and CPU architecture	Linux with an x86 CPU architecture or the Solaris OS with a SPARC® processor
Windows manager	GNOME 2
Office productivity	StarOffice software suite
Internet browser	Mozilla 1.x
Mail and calendar client	Ximian Evolution
Windows emulator	WINE or Codeweaver's CrossOver Office
Integration clients	Citrix ICA, Tarantella AIP, and/or RDP

The potential benefits of a Linux desktop solution are:

- Lower TCO compared with Windows clients due to significantly lower software license costs
- Longer refresh cycles

  The Linux desktop solution can use lower specification hardware; therefore, it extends the life of the current estate.
- Improved performance compared with Windows desktops

The drawbacks of a Linux desktop solution are:

- Not all of the Windows applications are available to the user through WINE or RDP and ICA.
- Linux skills are be required for at least some IT staff, and users require retraining, even though the GNOME look and feel can be made very similar to Windows.

Macintosh Desktops

There is a significant interest in the maintenance of support for the Apple Macintosh product set, particularly among users of special graphical and publishing applications that are developed specifically for this platform. These users can be integrated into an overall desktop architecture in a number of ways:

- Mac OS 9 and OS X clients exist for UNIX® printing (for instance, the common UNIX printing system known as CUPS) and SAMBA and NFS file sharing (see "File Services" on page 21).

- A variety of Mac OS email clients are also available (for example, the native Apple mail client, Mozilla, Netscape™, or Eudora).
- Several office productivity suites exist for Apple users, including Microsoft Office and a port of OpenOffice 1.0 (the open source version of the StarOffice 6.0 software).

For more information, see:

- `http://www.apple.com/macosx/pdfs/MacOSX_for_UNIX_users_TB.pdf`
- `http://www.apple.com/downloads/macosx/unix_open_source/`

Personal Digital Assistants and Mobile Devices

The number of Personal Digital Assistants (PDAs) and mobile devices has proliferated considerably in the last four or five years. The options range from sophisticated mobile phones, such as the recently introduced third generation (3G) handsets, to full-featured PDAs that are, in effect, reduced-footprint PCs, complete with an operating system, productivity and business applications, and browser. Currently, there is no single standard operating system or software stack for such devices. The options available in the marketplace include:

- Java technology-based solutions

  Generally, these are mobile phones, PDAs, and some gaming devices. These operate the Java Virtual Machine (JVM™) software and a selection of services most likely corresponding to one of the profiles defined by the Java Community Process (for example, the Mobile Information Device Profile, or MIDP). These devices generally run a number of standard productivity applications such as simple word processors, spreadsheet applications, calendars, and email clients, and they are capable of provisioning and executing Java applications, such as games, on a *pay-per-use* basis.

- PDAs based on variants of the PALM operating system, such as Palm, Handspring, and Sony

  These devices tend to have more capacity than phone handsets and carry a larger range of built-in applications. Some (for instance, the latest Palm products) ship with the JVM software to facilitate dynamic application provisioning.

- PDAs that run on Win CE or PocketPC

  Basically, these are subsets of the Microsoft Windows operating system family that provide a potential platform for ports of the many Windows-based applications.

Given the wide variety of clients available, most IT organizations will probably look to identify a *least common denominator* solution that will enable the support for the widest variety of client devices. Some of the architecture decisions that designers need to examine are:

- Integration with office productivity suites

  Both Microsoft Office 11 and the StarOffice 7 software will provide compatibility with some, but not all, PDA office personal productivity applications.

- Email and data synchronization

  Emerging standards such as SyncML might assist in simplifying the options for achieving synchronization between handheld devices and enterprise mail message stores. For more information about SyncML, see: `http://www.openmobilealliance.org`

- Choice of integration software layer for delivery of server-based applications (for example, Tarantella, Citrix, or other)

  Citrix appears to support more native client types than Tarantella, but both products have the capability to deliver to an HTML browser, provided that an appropriate Java VM is present (see "Application Integration Solutions" on page 18).

UNIX Workstations

There is still a considerable installed base of technical workstations based on RISC processors and various flavors of UNIX. The SPARC workstation with the Solaris OS holds the greatest market share in this sector, but systems produced by Hewlett-Packard (PA-RISC / HP-UX), IBM (POWER / AIX), and Silicon Graphics (MIPS / IRIX) are still in use, as well as even older similar workstations. For many of these users, there is a requirement to interoperate with and use office productivity applications, Internet browsers, and other applications. In most cases, the only feasible option is to install open source equivalents, as most of these have been ported to the more popular variations of UNIX. Finally, Tarantella and Citrix clients exist for most UNIX variations. These modules can be used to deliver applications that are served from a Windows system.

Application Integration Solutions

Windows Terminal Services (WTS) is a facility that was first provided by Microsoft as an add-on to Windows NT 4, and subsequently, it has been delivered as an inherent component of Windows server operating systems (Windows 2000 Server and Windows 2003 Server). WTS delivers a presentation layer service to suitably equipped client devices through Microsoft's RDP. In this context, a suitably equipped client could be a PC running Windows ME or XP, or it could be a system that emulates the RDP client such as Tarantella Enterprise 3 or WinConnect S.

A variant solution used by many organizations is to install Citrix MetaFrame on the Windows server to deliver applications to devices, such as a Windows-based terminal or a traditional desktop PC with Citrix client software. Citrix MetaFrame

runs on a Windows server in parallel with the WTS functionality, but it has some lower level *hooks* into the operating system that bypass WTS and overcome some of its limitations. Regardless of whether WTS, RDP, or Citrix is the delivery mechanism of choice, the user will have to pay for additional Microsoft licenses, known as Client Access Licenses, for each concurrent client device or system.

There are several contenders in the marketplace that can be used to deliver Microsoft Windows applications to heterogeneous client systems (for example, Tarantella Enterprise 3, Citrix MetaFrame, ThinSoft's WinConnect, and Rdesktop). The following table compares the attributes of these products.

TABLE 3 Windows Terminal Services Software Comparison

Attribute	ThinSoft	Tarantella	Citrix	Rdesktop
RDP version supported	5.2	4.0	N/A	4.0
Max Colors*	16 million	256	16 million	256
Sound Support*	Yes	No	Yes	No
Local printer support	Yes	Yes	Yes	No
Local disk access	Yes	No	Yes	No
Local communication port access	Yes	No	Yes	No
Requires middleware†	No	Yes	Yes	No
Client GUI	Yes	Yes	Yes	No
Support for UNIX servers	No	Yes	No	No
Support for z/OS and OS/400	No	Yes	No	No
Support for browser access	No	Yes	Yes	No
Command line options	Yes	Yes	Yes	Yes
Technical support	Yes	Yes	Yes	No
Requires MS licenses	Yes	Yes	Yes	Yes
Cost per client‡	$99.00	$250.00	$349.00	Free

* Maxcolors, sound, and disk access require RDP 5.2 (W2k3 and Windows XP).

† Middleware is defined as the addition of an extra software product, either on the host system (for instance, Citrix), or installed on an extra server layer (for example, Tarantella).

‡ MetaFrame XPe

RDP 5.0 and 4.0 do not support sound or 24-bit color. This limitation also applies to Tarantella (which relies on RDP), but it is overcome by Citrix, which bypasses the RDP mechanism to deliver these features to its clients. This situation changed with the release of Windows 2003 Server and RDP 5.2. These new products mean that RDP clients are able to provide the features that most users need.

In summary, users with mixed Windows, UNIX, and Linux environments should evaluate Tarantella and Rdesktop. Users who primarily access Windows applications should evaluate Citrix and WinConnect S.

When considering a possible move to thin-client solutions based on WTS and RDP or Citrix ICA, it is important to understand that there is a considerable amount of work involved in ensuring that the organization's many (typically, more than 1,000) Windows applications can successfully be delivered through the proposed integration layer. Generally speaking, Windows applications fall into three main categories:

- Applications such as Microsoft Office and terminal emulators that are designed to work with or are already known to be compatible with WTS.

  These applications can be delivered by using Tarantella or Citrix with no further effort.

- Applications that have not been integrated and/or tested

  Many applications are potentially WTS compatible, but they have not been integrated and tested in such a configuration. Allow at least two to three days for each application.

- Applications that cannot be integrated with WTS

  Some applications—thankfully, a minority—can never be integrated with WTS, particularly old 16-bit programs and software that assumes that it has sole ownership of the hardware. For these applications, the obvious choices are retirement, replacement, or relocation to a *legacy* fat-client system that will be retained and supported for a limited user population.

For thin-client delivery, additional design considerations are the allocation of applications to servers and the location of servers. There is an obvious design and cost trade-off between colocating all of the application servers into one or two data centers (which requires adequate network bandwidth), as opposed to a wider set of locations. Within the data center, there is also a set of decisions that need to be made about how applications are to be distributed. For instance, should each server provide all of the applications, or will groups of servers specialize in *advertising* sets of application services?

A final consideration for service delivery to all clients is the degree of integration between presentation layer software products and portal servers. For example, Tarantella is closely integrated with the Sun™ ONE Portal Server software, such that Tarantella-mediated applications can be built into and selected from a portal generated home page.

A Tarantella, Solaris OS, and StarOffice software benchmark is available at:

`http://www.tarantella.com/whitepapers`

You should also read the following Sun BluePrints OnLine articles:

- "Sizing Sun Ray Servers Running Windows Applications with SunPCi IIpro Coprocessor Cards" (November 2001)
- "Supporting Microsoft Windows 2000 Server Applications from Sun Enterprise Servers" (June 2001)

Services and Servers

This section looks at the variety of services that should be provided to the desktop user and the servers that can provide them. For each service, some of the architectural requirements relating to performance and resilience is also examined.

File Services

There are two main methods by which file storage services can be provided to users of thin (or thick) client systems: Network File System (NFS) and Common Internet File System (CIFS). NFS is a long established protocol used by UNIX systems to distribute file systems across a network, enabling users of heterogeneous clients to remotely mount directories (subject to security settings) from a file server, as though they were components of the local file system. CIFS is the latest of Microsoft's file networking protocols and is used primarily by Windows clients to mount server file systems as additional *drives* on their own system.

Both solutions can be provided and supported by Solaris OS and Linux servers. NFS is a standard operating system component for these servers, and CIFS can be delivered by an open source project known as SAMBA (the name derives from Server Message Block—the underlying protocol used by CIFS and its predecessors). The decision as to which file service to employ is partly dependent on the organization's familiarity with the underlying technologies and the requirements of their users. Some pointers are outlined below:

- UNIX and Linux clients can access both NFS servers and SAMBA servers (with `smbclient`).
- Windows clients will find it easier to access SAMBA servers, although Windows-based NFS clients are available.
- Users who have implemented Active Directory (see "User Authentication Services on page 23) will be able to integrate the latest version of SAMBA 3.*x* into an Active Directory (AD) forest as a domain server. NFS servers will not be recognized by AD.

CPU requirements for NFS file severs are not typically onerous. One estimate suggests that a typical user generates one to two file system events per second and consumes on average around 0.2 MHz of CPU capacity. A 400-MHz CPU should, in principle, be able to support 2,000 concurrent clients. Memory requirements are also fairly minimal. Around 1 megabyte per concurrent user will suffice unless particularly large data volumes are involved. A more significant issue is the bandwidth required.

Bandwidth needs to be sized so that the peak data volumes expected can be delivered with some margin of safety. For example, if each of 1,000 users will generate 40 megabytes of data traffic during the peak hour of the day, you should configure at least two 100-megabits per second connections to ensure an ample safety margin (the average throughput will be 90 megabits per second with a peak of around 210 megabits per second). It is important to ensure that sufficient disk capacity is provided in a resilient format, such as RAID 1 or RAID 5, and that this disk space is split across a reasonable number of spindles for improved I/O performance.

Print Services

Print services are typically provided by the same software and the same servers as file services. There is a choice of printer hardware configuration, with four main solutions being available in most cases:

- Linux and Solaris OS printers

  The queues for these printers are managed by the server to which they are attached.

- Windows printers

  Many organizations prefer to have their printers attached to Windows servers, even where the majority of their applications are not Windows-based. In this case, the print queues are on the Windows server.

- LAN printers

  Some printers have built-in networking capability and manageable elements. They can exist as a *standalone* device. In this case, the associated printer queues can be on any server and might exist on several servers concurrently.

- Sun Ray ultra-thin client devices

  Printers can be attached to the USB ports found on Sun Ray ultra-thin client device. The print queue for such printers is on the Sun Ray ultra-thin client server, which currently holds the session for the device.

The selection of the printer configuration is mainly a matter of individual choice. At one time, there was some preference for Windows printer servers because of a shortage of printer drivers for other platforms. This situation has changed, however, with the advent of networked printers and the greater availability of drivers. Some

of the guidelines for file services are also applicable to print servers. Most notably is that UNIX and Linux clients will have some difficulty accessing Windows-attached printers.

Administration of printers is often a time-consuming task for IT organizations. It is advisable to try to ensure consistency in the naming of printers and printer queues across the organizations. For the Solaris OS-attached servers, it is possible to create scripts that can identify the physically closest printer automatically.

Limited information is available on how to size print servers as distinct from file servers. Probably the most important consideration is the locality of the server with respect to the queue of the printer the server holds. It is generally not recommended to attach high-volume graphical printers to a server by using a low-bandwidth WAN connection.

An additional consideration is the amount of disk space required for holding spool files (that is, queued files awaiting output to a physical printer). The required space will be implementation-dependent and will be significant if many large print files are produced or if printers are removed from service (for example, for maintenance).

User Authentication Services

There are at least four currently available and widely used services that enable users to authenticate themselves and that can be used to give or withhold access to system resources, applications, other services, and physical devices:

- Lightweight Directory Access Protocol (LDAP)

  LDAP is an open standard (defined by IETF RFC 3377) that is widely found on UNIX and Linux systems.

- NT domain controllers

  These controllers were introduced as components of Windows NT. Each primary domain controller (PDC) holds authentication details (for example, name and password) for resources in its domain and might enter into trust relationships with other domain controllers. This solution is still widely used, as many users have not yet decided when or whether to upgrade to AD.

- Active Directory

  Active Directory (AD) was first introduced with Windows 2000 Server and is further enhanced in Windows 2003 Server. It extends the concept of domains into a *forest* of security *trees* that have a central repository of authentication data, but can delegate some administrative tasks to lower level controllers. AD claims to be LDAP-compatible, but the LDAP integration with AD, as shipped in Windows 2000, presented some challenges. With the release of Windows Server 2003, LDAP integration with AD has been improved.

- Novell directory service

  Novell directory service (NDS) has been in existence in various forms for many years. It is not directly compatible with LDAP or AD, but it can be integrated with either of them with some difficulty. NDS still has a significant user base, but most users anticipate moving to AD or LDAP in the future.

The authentication service is a difficult choice for organizations. In general, users of NDS and NT domain services are looking to migrate to LDAP or AD because they recognize that their current solutions will not be supported indefinitely. Those enterprises that plan to participate in future Web services delivery will also need to build software stacks based on a more contemporary identity services solution.

The move to either AD or LDAP is not without some difficulty because there is a need to migrate user and resource details to the new solution. As a general recommendation, the needs of users who have mainly (or solely) Windows application servers and are committed to the Microsoft proprietary .NET stack will be best served by migrating to, or continuing to use, the AD service. Other users will be better enabled to retain flexibility and openness by the adoption, or continued use, of LDAP-based authentication services.

For users who have both AD and LDAP solutions, there are several meta-directory products that can permit two-way integration between these two authentication servers.

Mail and Messaging Services

In recent years, there has been rapid growth in the use of email and, more recently, instant messaging services. These services are typically provided by a client-server software combination. The possible client solutions (which is discussed in “Software Components on page 27) communicate with servers that handle message storage and transfer by using one of several standard protocols such as IMAP4, POP3, webmail, or the proprietary MAPI interface, which is specific to Microsoft environments. Although Microsoft Exchange (which supports MAPI clients) has the largest share of the corporate email market, it offers a number of challenges for users who plan to evolve with the product:

- Users who upgrade to Exchange Server 2000 also have to implement AD services—a reasonably complex project for most organizations.
- Exchange users are locked into a fairly expensive and proprietary software product with costly upgrade plans.
- Exchange architectures generally involve a proliferation of small or medium servers that have a relatively high TCO because of their associated management costs.

A number of alternative solutions exist, including IBM Notes, Openwave Mail, Oracle Collaboration Suite, and Sun™ ONE Messaging Server. A viable candidate product will have most, or all, of the following characteristics:

- Support for HTML (webmail), IMAP4, POP3, and LDAP open standards
- Integration with a selection of antivirus and SPAM prevention products and significant built-in SPAM prevention mechanisms
- Integration with a variety of voice mail systems to provide a unified messaging solution
- Support for Simple Authentication and Security Layer (SASL), which permits the use of several different authentication mechanisms through an extensible security interface
- Extensibility and manageability, along with significantly lower TCO than Microsoft Exchange
- Support for instant messaging for users who need to implement an internal message exchange for conferences, alerts, chat, and news, as opposed to depending on a public network such as AOL or Yahoo

Calendar Services

The next web-based interactive application expected to become widespread is the calendar. Like email, the *e-calendar* will become another ubiquitous collaborative application, as people experience the benefits of sharing calendar information and using online interactive scheduling to manage their time, coordinate with family and friends, and track changes to work schedules. The recent ratification of new Internet standards for calendars has been a catalyst in moving calendars and scheduling into mainstream Internet activities. Today, iCalendar (RFC 2445) describes a standard schema for calendar objects on the Internet. The iTIP standard (RFC 2446) describes how to use iCalendar objects to perform common scheduling tasks such as publishing an event or inviting someone to a meeting. The iMIP standard (RFC 2447) describes how the iTIP scheduling operations can be performed over email. The CAP standard (which is work in progress by the Calendaring and Scheduling Working Group of the Internet Engineering Task Force) will provide a real-time access protocol for client applications to talk with calendar servers.

Sun Ray Ultra-Thin Client Servers

Sun Ray ultra-thin client servers are Solaris OS-based systems that run the Sun Ray ultra-thin client server software. This software is currently in its second major revision (2.0) and provides support for Hot Desk Protocol, which delivers frame buffers to Sun Ray ultra-thin client devices and transmits mouse clicks and

keystrokes to the servers. These servers are essential to the operation of Sun Ray ultra-thin client devices because they provide the CPU and memory resources for user sessions.

Version 2.0 of the server software enables the Sun Ray ultra-thin client device to coexist in networks that have other DHCP servers present. Version 1.3 of the server software could not coexist in such a network. Version 2.0 also provides for the optional encryption of data traveling between the server and the client. The key for this encryption is currently hard-wired into the server software.

There is no simple guideline for the ratio between Sun Ray ultra-thin client devices and server resources required to support them. Guidelines provided by the Sun Ray ultra-thin client engineering group suggest that each UltraSPARC® III CPU can support 25 concurrent Sun Ray ultra-thin clients. Such a guideline, however, ignores other software that is running on the server and user interaction with the server. In a real world situation, the typical user might be running a Tarantella or Citrix native client to present Windows applications and might also be accessing an Internet browser and an office productivity tool (for example, the StarOffice Office Suite), running on the same or a different server. Experience in the field suggests that some of these tools might require frequent screen refreshes caused by user interaction or, in the case of the Internet browser, caused by Shockwave animations or streaming video. Some additional guidelines to consider are:

- Always use a server with at least two CPUs whenever possible.

  This avoids the situation in which a single *thread* spins on a CPU and *locks out* other Sun Ray ultra-thin clients. A *thread* is a section of executable code that is recognized by the operating system as a *executable* entity in its own right.

- Base a pilot or proof of concept implementation on the server platform that is expected to be used in a more complete roll-out.

  This enables you to determine the resources required per user more accurately, assuming that the application set under test is representative of the production environment.

- If the users are going to access the Internet from their devices, consider installing a copy of the browser on each Sun Ray ultra-thin client server.

  This might reduce the additional traffic and processing overhead caused by users browsing dynamic pages through an intermediate layer such as Tarantella or Citrix.

Memory usage per user will vary depending on the applications in use. General figures vary between 40 and 100 megabytes per user, in addition to 64 megabytes shared system memory. In a typical Sun Ray ultra-thin client server configuration, the operating system is mirrored across two disks. Sufficient swap space should be provided to allow for inactive sessions. The quantity of the inactive sessions is dependent on user work patterns and Sun Ray ultra-thin client server software security settings. The inactive sessions will have a footprint of 40 to 100 megabytes each.

Software Components

This section covers the various software components that make up the software stack for most users. It identifies alternative products and solutions that are available and gives some pointers on selecting appropriate products.

Office Productivity

Although Microsoft Office has been the market leader for a number of years, there has been a significant user reaction to recent changes in Microsoft's licensing policies that appear to be moving toward the concept of software rental and, at the same time, to be forcing users to update their products on a frequent basis. In this environment, the advent of Sun's StarOffice Office Suite (and the associated OpenOffice open source project) has been welcomed by many organizations.

A feature set comparison of Microsoft Office and StarOffice is online at:

`http://wwws.sun.com/software/star/staroffice/6.0/compare/`

StarOffice and OpenOffice are now viable alternatives for organizations that are cost-sensitive and/or that do not want to participate in Microsoft's Software Assurance program. The main benefits of StarOffice 6.1 (OpenOffice 1.1) are:

- Low licensing and support costs
- Improved Microsoft file format compatibility
- Native XML file formats
- PDF export facility

Users who might be less suited to StarOffice might choose to stay with Microsoft Office for the following reasons:

- Large numbers of embedded VBA macros (although these might need to be modified for compatibility with future Microsoft Office releases)
- Significant presence of ISV applications that integrate with Microsoft Office
- Frequent interchange of documents with external Microsoft Office users

No sizing data is currently available for StarOffice served from a UNIX or Linux server, but a sizing guide is expected some time in the future. The minimum supported configurations for StarOffice clients is online at:

`http://wwws.sun.com/software/star/staroffice/6.0/techfaq.html`

Internet Browser

Traditionally, two contenders for the role of Internet or Intranet browser have been available: Internet Explorer and Netscape Communicator. A number of other contenders (for example, Mozilla, Opera, and Safari) have appeared recently, however, of which the leading offering is possibly the Mozilla open source project (on which Netscape 7 is based). There are also issues surrounding browser support on mobile devices—many of which are not based on the Microsoft operating software.

For home users and users who need access to a wide variety of multimedia content from the Internet, Internet Explorer remains the most viable choice due to its support of ActiveX controls and integration with a large number of plug-ins. However, there is still a strong linkage between Internet Explorer and the underlying Windows operating system, such that users need to upgrade to later versions of Windows to take advantage of future browser enhancements (other than the frequent security fixes issued by Microsoft).

Users who have limited Internet requirements, who are using a browser primarily for business purposes, or who have a mixed desktop client environment (which appears to be a growing community), might be better suited to the Mozilla or Netscape product, which also provide an open source mail client.

Window Managers

The choice of a window manager is almost a religious or philosophical debate among the more knowledgeable. For other users, it is a matter of indifference (or zero choice). The four main contemporary contenders are:

- Windows XP (and its predecessors and successors)

  Generally, Windows XP is agreed to be friendly and usable. As a consequence of its widespread use among home and educational users, it is understood by a very wide audience.

- CDE

  CDE was the default desktop environment for the Solaris OS (and therefore for the Sun Ray ultra-thin client). It has a slightly old-fashioned look and feel, and it is likely to be displaced by its successors: KDE and GNOME 2.

- KDE

  KDE is becoming the window manager of choice for more technically oriented Linux users due to its more *modern* look and feel, flexibility, and range of configuration options.

- GNOME 2

  GNOME 2 is the latest version of an open source project. It is not as flexible as KDE, but it has attractive graphics and can be configured with a *skin*, which gives it the appearance (and functionality) of Windows XP. GNOME 2 is probably the best choice for users with XP experience who are migrating to UNIX or Linux platforms.

Mail and Calendar Clients

As with other elements of the software stack, the choice of mail and calendar clients depends on the user's system environment and acceptance of open source software.

The choice is essentially between a Microsoft *defacto* standard and open source equivalents. The main contenders are:

- Outlook and Outlook Express

  These clients were designed to integrate with Exchange servers that provide both mail and calendar services. They have been targeted by many email-attached viruses that have been able to exploit a variety of security vulnerabilities.

- Ximian Evolution

  This client is based on an open source project and integrates with a number of backends (for example, Exchange or IMAP and iCalendar-based servers, such as the Sun ONE Messaging Server and Sun™ ONE Calendar Server software). Ximian Express has a look and feel very similar to Outlook, and it incorporates Personal Information Manager (PIM) functionality (as does the full Outlook product). Ximian Evolution is not supported on Windows platforms. It runs only on UNIX or Linux platforms.

- Weblicon Calmena

  This client runs on a variety of platforms (for example, Windows, UNIX, and Linux) and integrates with either Exchange or IMAP4 and iCalendar servers. Weblicon Calmena has more limited functionality than Evolution and does not emulate the appearance of Outlook. However, it is a reasonable choice for organizations that want to displace Microsoft totally and that have a variety of client platforms.

- Mozilla and Netscape

  These open source clients provide a useful mail client that integrates with IMAP4 servers and provides limited folder management. They are available for most operating platforms.

Nonfunctional Requirements

This section outlines a number of additional topics that are of importance in the design of desktop (and other) architectures and that are collectively referred to as nonfunctional requirements.

Security

A wide-ranging discussion on security topics is outside the scope of this article. For more information about this subject, see the selection of Sun BluePrints articles under the Security heading at:

`http://www.sun.com/solutions/blueprints/`

There are, however, a number of security issues that are particularly relevant to desktop environments.

Virus Infection

Virus infection is an issue that has had (and probably will continue to have) an impact on every IT user. At present, most viruses are targeted at a specific operating environment (Microsoft Windows), associated email client, and browser. Although in fairness, it should be pointed out that this tendency is most likely due to a combination of the wide availability and usage of these products. There are also potential architectural weaknesses in the design and implementation of some software products that make them easier to attack than others. In any event, organizations need to take comprehensive precautions against viruses. A program of defense should include:

- User education
- Hardening of server systems by removal of unnecessary subsystems, services, and accounts
- *Lock down* or restriction of the capabilities of client devices
- Implementation of virus scanning and spam filtering software on mail servers and/or mail relays
- Implementation of antivirus software on vulnerable operating system platforms (for example, Windows servers and clients)
- Some of these measures are more effective and/or more easily implemented in a thin-client environment. This can reasonably be viewed as an argument in favor of thin-client solutions in terms of both vulnerability and TCO.

Federated Identity

The topic of user authentication was covered in “Services and Servers” on page 21. There is an additional architectural debate that some organizations need to follow and make a decision on, which is the selection of standards and products for the support of federated identity. This topic is important to organizations that are seeking to build communities of external users (for example, customers and suppliers), but it is also important to organizations that are building portals to support a mobile workforce. The main protagonists in this debate are the Liberty Alliance (`www.projectliberty.org`) and Microsoft with their Passport product.

User Identification

A number of current and forthcoming software and hardware products are targeted at increasing security against *spam* (bulk unsolicited email) and at preventing unauthorized access to applications and resources, along with the authentication services described in “Services and Servers” on page 21.

Some examples are:

- Smart cards, such as those based on the Java Card technology

  Smart cards can be used with readers to provide another security *leg*. Users need to present a valid smart card in conjunction with the correct user name and password to gain access to the system(s).

- One-time password or token generators, such as SecureID and SafeWord

  Token generators can be used in conjunction with a memorized PIN to generate a one-time key that is the correct response to a challenge issued by server software (provided by the company that manufactures the token card). This mechanism is often used to provide authentication for remote users who want to gain access to an internal network by using a dial-up connection or a VPN.

- Biometric devices

  These are relatively expensive solutions that aim to provide authentication based on unique personal identifiers such as a thumbprint or a retinal pattern. Initially used for enhanced physical security (for example, access to buildings and data centers), they are now being used to extend to include controlled access to secure systems such as the defense and financial networks.

Single Sign-On

Single sign-on is a facility whereby the user is able to pass through one authentication gateway and be granted automatic access to all of the applications to which they have been given access rights. This facility can be provided today by a combination of authentication services with scripting and configuration set-up in a portal server or integration software layer.

Eventually, most applications and services will be integrated with an authentication server and undertake their own look-up with the server, eliminating the need for *below the covers* scripting solutions.

Data Security

Data security, as an aspect of the desktop architecture, should not be overlooked. A full discussion of this topic is outside the scope of this article, but a couple of key points need to be remembered during system design:

- Security of data storage devices

  Consider the deployment of resilient storage devices such as RAID 1 or RAID 5 disk arrays.

- Backup and recovery of data

  Consider the addition of backup devices (for example, tape libraries) in conjunction with appropriate software and backup regimes.

Manageability

Manageability is another topic that is covered well by many other documents, but it also constitutes a vital component of any desktop solution. Some considerations that are specific to the desktop environment include:

- Sun Ray ultra-thin client administration

  The facilities provided by the Sun Ray ultra-thin client server software to manage the clients generally needs to be augmented by user-written scripts that provide additional features, such as printer management and session management.

- Sun Ray ultra-thin client and Solaris OS servers (for example, directory, file, print, and messaging servers)

  These servers can be administered by using the Sun™ Management Center software. Generally, the designer needs to provide a separate management LAN and also consider the automation of frequent management tasks and integration with *manager-of-manager* tools, such as HP OpenView, Unicenter TNG, and Tivoli.

Availability and Reliability

Some aspects of resilience have particular relevance to desktop systems design (for instance, Sun Ray ultra-thin client server resilience). The Sun Ray ultra-thin client server software provides failover between two or more servers in a logical cluster (that is, a user session on a failed server is lost, but it can be re-established on a second server, if available). The software also provides for load balancing between servers so that user logins are distributed on a *least heavily loaded* basis when multiple servers are available.

Tarantella and Citrix provide similar failover and load balancing facilities in their integration software. In combination with Sun Ray ultra-thin client server software, users with failed sessions can reattach to their existing Citrix or Tarantella session after they have reconnected to an operational server.

Other service types also have their own means of ensuring that availability goals are met. Authentication services such as LDAP or AD generally support the concept of backup servers that can hold active or nonactive copies of user identification and attribute data.

File and print services can be based on cluster solutions such as the Sun™ Cluster 3.0 software, which ensures that service can be re-established following a server failure. Storage continuity also needs to be provided to ensure complete continuity.

Messaging and calendar software products are generally designed with a built-in concept of horizontal scaling (for example, user mailboxes can be held on a single mail server or spread across several servers). In this environment, failure of a single server causes partial loss of overall service, but complete loss for some users. Designers might want to consider a cluster solution if a higher level of availability is required.

Vertical and Horizontal Scalability

Most of the services discussed in this article can be loosely (that is, logically) clustered. Therefore, designers can add capacity relatively simply to desktop server infrastructures by adding more purpose-built servers (for example, Tarantella servers or directory servers) to the existing infrastructure.

Although this gives the architect a straightforward solution for scaling the design, it is worth considering the trade-off between horizontal and vertical scaling. As the number of physical servers and operating system instances increase, so too does the requirement for system administrators, space, and power. Most UNIX-based services also scale vertically. For example, directory services and mail services achieve near linear performance improvements as more processors, memory, and I/O capacity are added to the system.

Network Design

Consider network bandwidth, latency, and resilience when determining the type and location of client devices, printers, and servers. It is important to recognize that some client devices (for example, the Sun Ray ultra-thin client devices) have fairly large bandwidth requirements (around 1 megabit per second) and need to be located at least on the same campus as their associated servers, if not on the same LAN segment, unless the organization has a particularly *fat* WAN and an aversion to distributed servers.

Other clients like those using Tarantella or Citrix have more limited network needs and can be successfully located at the end of a low-bandwidth pipe. Printers constitute an intermediate case. Low-volume printers require limited network capacity, but fast printers with high-resolution graphics can create significant network traffic.

Usability and Environmental Factors

There is considerable variation in the size of screen and display resolution between the smallest mobile phone and a high-resolution engineering workstation. You should consider how users will use the device (for example, the applications required and the frequency and duration of access).

The Sun Ray ultra-thin client and some other thin clients (for example, Windows-based terminals) have no moving parts; therefore, they generate almost no noise in operation. This feature can be particularly useful in crowded environments such as call centers and training rooms. In addition, Sun Ray ultra-thin clients require less than 20 watts (typically, 11 watts) to operate, as compared with conventional desktops that require 200 to 300 watts (excluding display devices in both cases). This can lead to a significant reduction in electricity costs, even when the Sun Ray ultra-thin client server power budgets are factored into the equation.

Given the lower power consumption of the Sun Ray ultra-thin client, it necessarily follows that the heat output, and therefore the cooling requirements, for these clients are also significantly lower than the heat output of fully functional PCs or workstations.

Implementation Roadmap

This section examines the likely steps that an organization would follow to design and implement a desktop solution that meets its business needs. It examines both the sequence and the content of the steps most users go through to arrive at a best fit solution.

Initial Evaluation

The best way to understand many of the aspects of current desktop technology (for example, the Sun Ray ultra-thin client environment) is to see them in action. Most potential users will benefit from a demonstration and/or reference visit to become familiar with the products and architectures under consideration.

An important parallel activity is to evaluate the likely cost savings and return on investment (ROI) that can be achieved by migrating to the proposed architecture. A useful tool, originally written on behalf of Citrix, that can be used to estimate the cost savings for thin-client solutions is online at:

`http://www.acecostanalyzer.com/`

Sun Professional Services (SunPS℠ program) and Sun's partner organizations can provide services to assist customers in assessing ROI and building the business case for a more efficient desktop architecture.

Requirements Definition

Defining the requirements generally involves understanding the current environment, the user, and the organizational needs that drive the architecture. Attention should be paid to:

- The characteristics of the current desktop environment (features, benefits, and drawbacks) from the perspective of users, managers, and system administrators.
- Current architecture limitations (for example, network bandwidth, support issues, software, and hardware obsolescence)
- Future technology changes (for example, availability of mobile devices, increased bandwidth, and wireless networks)
- Future (planned) changes to user groups, working practices, and scale and location of operations

- Possible integration requirements with partners (for example, suppliers, customers, and peer organizations)
- The probable budget and time scale for a migration project
- Any other risks and constraints that might be applicable to the potential project

At this point, you should be able to *rough out* the architecture that will underlie the pilot phase and reconfirm the potential ROI with budgetary costs.

Pilot

Following the gathering of requirements, you should evaluate the technical feasibility of the *rough cut* architecture. The steps required in carrying out a successful pilot are:

- Identify key objectives and critical success factors (CSFs) for the pilot.
- Design a pilot architecture based on the output of the requirements definition phase. You should plan to use hardware components that are likely to form part of the eventual solution. This will help to confirm sizing requirements during the architecture phase.
- Plan to support users during the pilot phase. You should include time and resources for user training and incremental help desk or floor-walking support during the early days of pilot implementation.
- Run the pilot for a realistic period of time with a set of real users, employing the pilot solution as their main (preferably only) desktop environment. If the set of users is too small or if the period of time is too short, you will not generate useful results.
- Regularly review pilot progress, and at the planned end date, evaluate the outcomes and lessons learned from the pilot, including potential technical risks and possible contingencies or workarounds that might be employed in a full-scale rollout.

Full-Scale Architecture

Following the pilot, you should be able to progress to a full-scale architecture design. The steps in accomplishing this task are:

- Reconfirm the findings of the requirements definition phase.
- Understand the outcomes of the pilot phase.
- Use these inputs, together with available design and sizing documentation (such as this article), to build a high-level design.

- Review the design with users, managers, and support personnel for functionality, feasibility, cost, supportability, and technical risk.
- From the high-level design, derive build and configuration specifications for all of the components to be installed or repurposed during the rollout.

Implementation

Implementation details are specific to the environment, design, and technology. Apply the following general principles to the rollout project:

- Plan all phases of the rollout to minimize disruption to any given user group at any given time.
- Plan for incremental training and support for users going through the transition phases.
- Make backup copies of all user and system data prior to any significant change.
- Ensure that there is a back-out plan for major changes, especially those steps that have significant time constraints (for example, servers that are to be installed and commissioned over a weekend).

As mentioned before, the SunPS program and Sun partners can assist with the delivery of all of the phases described in this implementation sequence.

Third-Party URLs

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

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

Acknowledgements

The author would like to recognize Malcolm Verrall and Paul Shore for their contributions to the development of this article. He would also like to recognize Craig Bender and Don DeVitt for their comments on the article.

About the Author

Howard Carlton is an IT Architect with over 20 years of experience in the design and implementation of complex information systems. Since joining Sun Microsystems in early 2000, Howard has worked on a number of customer projects, including the migration of financial systems from Sequent systems to Sun Fire™ midframe servers and the design of a global supply chain management solution for a semiconductor company.

For the last 18 months, Howard has focused on the desktop solutions and has been leading a SunPS team that has architected and delivered a number of working examples of thin client computing, based on the Sun Ray ultra-thin client and StarOffice Office Suite software. This experience has been captured as best practice examples and incorporated in reusable service definitions, as well as architectural patterns, as demonstrated in this article.

Ordering Sun Documents

The SunDocs℠ program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals through this program.

Accessing Sun Documentation Online

The `docs.sun.com` web site enables you to access Sun technical documentation online. You can browse the `docs.sun.com` archive or search for a specific book title or subject. The URL is `http://docs.sun.com/`

To reference Sun BluePrints OnLine articles, visit the Sun BluePrints OnLine Web site at: `http://www.sun.com/blueprints/online.html`