

# A Comparative Discussion of Sun™ Secure Global Desktop Software and SSL VPN Technologies

White Paper

November 2006

## Table of Contents

Introduction.....	3
Sun Secure Global Desktop Software Architecture.....	4
SSL VPN Architecture.....	6
Technology Comparison.....	7
Protocol Isolation.....	7
Application Consistency and Cross Platform Client Support.....	8
Making The Right Choice.....	9
Sun Secure Global Desktop Software.....	10
Related Links.....	11

## Introduction

Organizations need to provide secure access to critical data and systems from a wide variety of client devices and environments. The classic model of users sitting at dedicated workspaces using the same computer day after day is outdated and does not reflect the current trends in today's fast moving business environment. Users are working from more locations and are required to access more real-time data than ever before.

There are several models for providing secure remote access to applications and data. Two of the most common are server-based computing and virtual private networking (VPN). In the last several years, a variation of the VPN model that relies on Secure Sockets Layer (or SSL) based VPN hardware devices has become popular in enterprise deployments. This document will focus on these SSL-based VPN systems and will not contemplate the more traditional VPN implementations.

Often, server-based computing products and SSL VPN devices are evaluated with the purpose of solving the same or similar sets of problems. While the overall design goals of each type of product are often similar, the two models differ radically in implementation and the overall user experience for end users. This paper will discuss the architecture of Sun's server-based computing product Sun™ Secure Global Desktop Software and how it compares to SSL VPN technologies.

As you read this document it is important to consider that Sun Secure Global Desktop Software was designed with the following goals in mind:

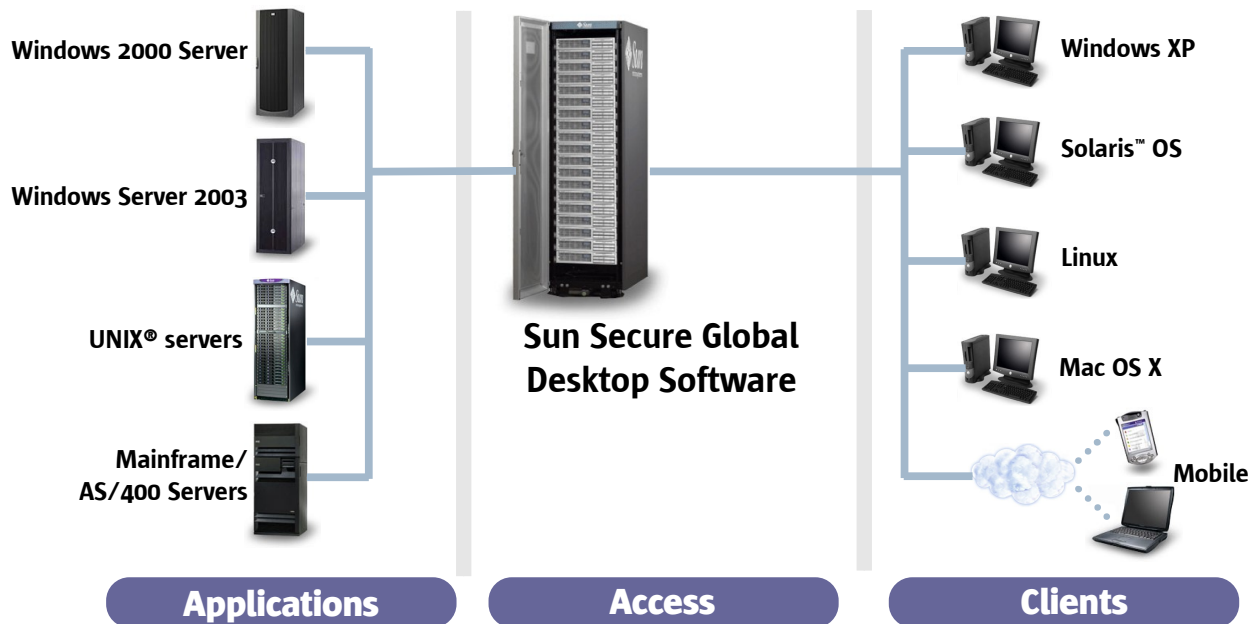
- Provide universal access to virtually all applications from nearly any type of client device.
- Provide for user and session mobility while maintaining industry leading security.
- Provide a single solution that can provide secure access to applications without adding costly and difficult to maintain third party emulation software on each client device.
- Provide an application access platform that is easy to administer and does not require an exorbitant investment in maintenance and support costs.

It is assumed that the reader has a working knowledge of Sun Secure Global Desktop Software. Additional product information can be found in the Sun white paper “A Product Overview: Sun Secure Global Desktop Software” available at:

<http://www.sun.com/software/products/sgd/whitepapers/>

## Sun Secure Global Desktop Software Architecture

Sun Secure Global Desktop Software leverages a 3-tier architecture to provide access to virtually all desktop applications, including those that run on Microsoft Windows, UNIX®, Linux, Mainframe or midrange servers.



- **The Applications Tier** – Applications run on centrally managed servers. These servers can be machines running one operating system, or multiple operating systems through the use of virtualization technology.
- **The Access Tier** – Sun Secure Global Desktop Software is typically hosted on one or more dedicated Sun Solaris™ OS or Linux servers. The servers in the access tier serve as a gateway, allowing many different client devices to access applications running on the application tier servers.
- **The Client Tier** – Popular client devices such as Microsoft Windows PCs, UNIX or Linux workstations, Internet devices, and Windows Mobile-based PDAs, can access applications running in the application tier.

Sun Secure Global Desktop Software can provide application access either through direct integration with a user's Start or Launch menu or using a web browser. The web browser option provides the most direct comparison to SSL VPN technologies so this is the method that will be discussed here.

In order to access an application using Sun Secure Global Desktop Software and a web browser, a user follows these steps:

- A Java™ technology-enabled web browser is launched on the local device.
- The URL of a Sun Secure Global Desktop Server is entered into the browser.
- The user is prompted for authentication info. The user provided information is checked against several different login authorities (as defined by the administrator) including optional two-factor authentication systems.
- Once the user is authenticated, a list of applications is displayed in the web browser.
- The user clicks a link in the web browser to launch the application.
- The user interface of the application is remotely displayed on the user's system, while their keyboard input and mouse movements are sent back to the server. This gives the illusion that the application is running on the local system, but in reality, the user is simply manipulating a high quality series of images streaming from the application tier server.

Note in this model that it is not necessary to install application software on the client devices. Applications are installed by the IT organization on application tier servers and the user is simply opening a remote session to those back-end servers. This allows users to freely move from device to device and have consistent access to all of their applications and data. The connection to the Sun Secure Global Desktop Software server is an encrypted SSL connection with a choice of cipher suites, including AES 256 bit strong encryption.

## SSL VPN Architecture

SSL VPN systems are designed to allow remote users to establish a presence on a network from an access point outside that network. In the SSL VPN model, access is provided to the data systems that would be available to the user if they were sitting at a node inside the network, but gives them the flexibility to roam outside the domain of the network and still access data as if they were inside.

In a typical SSL VPN implementation, a web browser is used on the client side to establish a connection to the SSL VPN device. Once a user is authenticated, a secure “tunnel” is created between the user's system and the remote network, providing access to the services of the target network for the remote device. The implementation details of these systems varies across vendors and a discussion of the relative merits of each is beyond the scope of this document.

To access data with an SSL VPN, a user typically follows these steps:

- A web browser is launched on the local device. Depending on implementation, vendors have different browser requirements (sometimes a Java technology-enabled web browser is required, other times Windows-only ActiveX controls are used, etc.).
- The URL of the SSL VPN device is entered into the browser.
- The user is prompted for authentication info and is authenticated.
- A SSL tunnel is created between the user's computer and the inside network.
- The user launches a local application such as an email client or a database client and the application can access data repositories (email servers, database servers, etc.) on the inside network even though those data sources would not be accessible if the tunnel were not in place.

Note in this model that all application execution is done on the local client device and only the data that the application is accessing is sent over the secure connection.

## Technology Comparison

Both the server-based computing and SSL VPN models provide highly secure access to data, but there are important differences that may make one model more suitable for a particular environment than another. This section will discuss important distinctions between the two methodologies.

The most significant difference between server-based computing and SSL VPN solutions is where the client application executes. This factor dictates the type of data that is sent down the secure connection to the client. In the SSL VPN model, users run local copies of applications and access data sources on a protected remote network (this can be characterized as a “client-server” architecture). In the server-based computing model, applications are installed and run on back-end application tier servers on a protected network and are simply manipulated remotely by users (this can be characterized as a “display tunneling” architecture). The server-based computing model does not preclude the use of client-server applications – in this model, the “client” is simply a back-end application tier server on the protected network.

Although there are solutions that combine the two approaches, when deciding which technology to use for a given set of applications it is important to consider the following factors:

### Protocol Isolation

When running an application in client-server mode, the application has a direct connection to the data source it is manipulating. When these applications are installed on local systems and access to the data is provided by an SSL VPN, this direct connection is extended to a node outside the protected network. In many senses, the device at the end of the tunnel becomes a full fledged citizen of the protected network. This opens the possibility that a nefarious third party could install software on the client device that could potentially infiltrate the protected network, capturing or even manipulating the remote data from the client device. Due to the risk of surreptitiously installed applications wreaking havoc on a secure network, many SSL VPN solutions will thoroughly inspect a client system for viruses, key loggers, etc. before accepting a connection.

Server-based computing solutions encourage application execution in the data center and provide a visual representation of the application to the client device, thus eliminating this direct connection to the data. Some server-based computing solutions, including Sun Secure Global Desktop Software, increase security with the addition of a server that lies between the client devices and the servers executing the applications. This server in the middle acts as a bastion host, brokering the client side and the server side connections, but never allowing a direct connection to occur. Both the client side and the server-side protocol streams are terminated on the middle tier, limiting exposure for the critical back-end servers.

## Application Consistency and Cross Platform Client Support

Many applications offer cross-platform client side applications to access back-end data sources. These applications are often available for Microsoft Windows, Sun Solaris OS, Linux, or Macintosh OS X based systems. Additionally, some applications provide a web-based interface as well that supports many operating systems through a standard web browser. One of the reasons that many companies are tied to a specific operating system for client deployments is the lack of availability of these applications on various platforms and, when they are available, the inconsistencies in user interface on each platform. For example, the Windows version of a client application might be more full featured or better implemented than the Linux offering simply because the Windows client has been around for a longer period of time and is more mature. Or, a web-based interface may offer compatibility with many different operating systems (and the HTTP/HTTPS data that is accessed is easily tunneled by an SSL VPN), but the interface is unlikely to have the same rich set of features that the full fledged client application has.

Server-based computing offers a solution by providing cross-platform access to an application running on a specific back-end operating system. This methodology provides several advantages:

- The most full featured version of the client software can be deployed, not the version that supports the operating system that most users have installed.
- Users can be trained on one user interface regardless of their choice of client devices.
- IT manages only one version of the application.
- Client hardware can be swapped out or replaced, even with a device that runs a different operating system, without affecting the application experience for users.
- Users can move from device to device with a consistent interface whether they are on their usual work system, an Internet terminal in a hotel or airport, or a mobile device.

Also, organizations often run critical applications that were developed for a specific operating system, either by in-house developers or by a commercial software vendor. As time moves on and new operating systems are released, the applications may not be compatible and may need to be revised. However, it is often the case that the developers of in-house applications are no longer available, or it may be cost prohibitive to upgrade a commercial package to support a new operating system. Companies find themselves in the tenuous position of relying on a critical application but not being able to update their client devices. Or, even if the application is supported in the new operating system, thorough in-house testing must be done to ensure compatibility. This leads to organizations delaying updates on their desktop systems and potentially opens them up to security issues on their client devices.

With server-based computing, application tier servers can be dedicated to a specific operating system and used to host the applications in their intended environment. In this model, the client operating system has no impact on the operation of the applications. This means that updates can occur at a more natural rate on the client devices and new operating system roll outs do not need to be tied to support for a specific application on that operating system.

## Making The Right Choice

Both server-based computing and SSL VPN technologies have advantages in specific situations, but it is the behavior of the users that is the most influential factor in deciding which solution to deploy. The types of applications people use, how they use them, from where, and on what devices will ultimately decide which solution is correct for a given organization.

When choosing a solution, the following points should be considered:

- Applications are not installed locally with server-based computing solutions. This allows users to work on systems that have not been previously configured or those that do not support the execution environment for a given application. This improves compatibility over SSL VPNs which generally execute applications on the local client device and are limited to applications that run on that specific device and operating system. Also, many SSL VPN systems need to have intimate knowledge of the protocols they are tunneling (IMAP, HTTP, etc.) and do not work with all types of data.
- Many SSL VPN solutions support Windows clients only or offer reduced functionality on non-Windows systems. This limits the types of client devices that can be deployed across remote offices, used by mobile workers, etc.
- Server-based computing solutions offer robust application publishing and management. On the other hand, the goal of an SSL VPN is to provide a simple and secure way to access a specific network, and application deployment and management is handled using other techniques. In some cases, SSL VPNs integrate with server-based computing solutions to provide application management, which may increase complexity over using one approach by itself.
- Some SSL VPN solutions provide excellent client side inspection features. Surreptitiously installed applications on a remote node connected by an SSL VPN can wreak havoc on a secure network, so many SSL VPN solutions will thoroughly inspect a client system for viruses, trojan horses, key loggers, etc. before accepting a connection. Server-based computing solutions employ a generally less invasive connection to the remote network and generally do not have client interrogation features.
- Many server-based computing solutions offer integrated heterogeneous application support. Because applications are executing on back-end servers and only their displays are sent to the client, it is simple to mix and match applications originating on different operating systems and platforms and combine them on a single pane of glass on the client device with no additional software installation. In an SSL VPN environment, if access to a UNIX or legacy application is necessary, for example, then third party emulators may need to be installed and managed on each client device.
- Application sessions can be suspended on one system and easily resumed on another with server-based computing solutions. This allows users to easily shift from location to location and from device to device. Because applications are often installed locally with SSL VPNs, that model is generally more appropriate for users that are always operating on the same client devices.

- Running applications locally offers better performance for demanding graphics applications. There is some overhead associated with moving potentially large amounts of visual data between a client device and a back-end server, so SSL VPNs are usually the most appropriate choice in these scenarios. Note that the amount of data being manipulated has an impact on this as well, though. For example, manipulating 3D wire frames in real-time is likely a better choice for a local application and an SSL VPN because the amount of data sent over the wire is small and the demands of rendering that data are high. On the other hand, a textured 3D model that includes textures that update in real-time (weather data or traffic, for example) may actually perform better in the server-based computing environment because only the currently displayed portion of the model needs to be sent to the client instead of the entire texture map, especially on lower powered client devices.
- The data stream sent to the client in the server-based computing model is primarily image and control data, with no “raw data” coming down to the client. Because applications are executing on application servers, this eliminates the chance that an application will cache raw data locally and leave potentially large amounts of sensitive information on a client system. In the SSL VPN model, the raw data is piped into a local application which may save it on the client device.

### **Sun Secure Global Desktop Software**

In addition to the above points, Sun Secure Global Desktop Software offers the following notable features:

- Sun Secure Global Desktop Software provides users the ability to access server based applications without having to manually install additional software. Many other secure access solutions require installing either native applications (and often these applications are available only for Windows based PCs) or costly third party emulation software to access 3270, 5250 and UNIX applications on each client device.
- Sun Secure Global Desktop Software helps to increase platform independence. The architecture allows a wide variety of client devices and browsers to be used, whereas other solutions often require specifically Microsoft Windows 2000 / Windows XP and Internet Explorer to successfully operate. This ties the user to a specific desktop configuration and greatly diminishes client flexibility and mobility. Sun Secure Global Desktop Software allows users to access applications from multiple devices, including PCs or thin clients running Windows, Linux or Solaris OS and Windows Mobile-based PDAs, using most Java technology-enabled web browsers, while at the same time improving the isolation of critical back end systems, reducing exposure to outside threats.
- Sun Secure Global Desktop Software helps reduce the cost of software maintenance on client devices and extends the lives of existing desktops. The preferred client device for most other solutions is a fat client (PC) with locally installed applications. The use of a fat client paired with the SSL VPN data delivery model generally means that new applications need to be distributed to each client device and maintained individually. In the Sun Secure Global Desktop Software model, applications are installed in a central location and used by multiple users. This reduces time spent installing and updating software, an important factor to consider as upgrades are released. It also allows users on under powered systems or those with older operating systems to access applications that would not run directly on their client devices, allowing older PCs to be used for longer periods of time.

**Related Links**

See the following resources for further information:

For the latest information about Sun Secure Global Desktop Software , see the website at <http://www.sun.com/software/products/sgd/>

Copyright © 2006 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A.

All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Sun, Sun Microsystems, the Sun logo, 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 architecture developed by Sun Microsystems, Inc.

UNIX is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, Ltd.

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a). 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 HELD TO BE LEGALLY INVALID.