

# Desktop Application Virtualization and Application Streaming: Function and Security Benefits

Tom Olzak  
August 2007

Current security issues caused by an increasing number of threats, application vulnerabilities, and the growing complexity of many computing environments is making traditional approaches to system maintenance and support very difficult—or impossible—to manage. Even the best attempts to maintain up-to-date endpoint device environments—including operating system and application security patch rollouts—often fall short due to cost or logistics issues.

Exacerbating the problem of endpoint management are the user demands for freedom to install their favorite productivity applications or use the latest handheld personal data assistant (PDA). In addition to the potential for a business network malware pandemic, incompatibilities between personal applications and critical business client functionality can result in business continuity challenges.

In this paper, I examine the challenges facing managers as they attempt to provide secure, continuous processing capabilities on their endpoint devices, specifically fat clients. This is followed by a look at where we've been, how we arrived at the current processing environment, and possible next steps—application streaming into virtualized user devices. I use Microsoft SoftGrid to walk through the process of setting up and managing this new approach to managing desktop and laptop security.

## Management Challenges

Today's IT managers are faced with a growing number of challenges when attempting to maintain end-user, fat client computers. The following list is taken from a ComputerWorld article by Galen Gruman (2006). The comments are mine.

- *Tracking who has what update.* As the number of applications increases across a growing number of end-user computers, many organizations cannot effectively determine which systems are up-to-date. This is a significant issue when the untracked updates are either security patches or mandatory changes to a critical business application.
- *Help desk costs.* Unless a standard image is deployed to all end-user systems, the number of help desk calls can be rather large. Even when standard configurations are used, the pressure to allow user-provided applications causes many companies to provide local administrator capabilities to laptops and desktops for installation and desktop configuration purposes. Conflicts between these changes and

- business critical applications can result in significant costs in both lost productivity as well as IT support.
- *Long security update cycles.* When vendors release security updates, the end-user device update cycle can be rather long. It includes testing to ensure nothing breaks followed by an often time consuming rollout; a rollout during which making sure every system is patched can be very difficult if not impossible.
  - *Actual usage tracking.* Licensing issues for large enterprise networks can be a career path for some unsuspecting member of Information Services. Understanding what applications are actually used, how often they are used, and who is using them is nearly impossible without special metering tools. Even then, traditional metering solutions usually require management of an agent on each system containing the software a manager wants to track. Finally, matching the number of licenses purchased to the number of instances of an application used is a matter of “close enough” guesswork.
  - *Incompatibility between application components.* In addition to potential conflicts when patches or end-user applications are installed, new applications mandated by management can also cause havoc when new or replaced operating system components cause failures in existing applications. Examples include dynamic link libraries (DLLs), Java versions, ActiveX controls, and other shared components or services. Add to these challenges the possible changes to registry controls, and you have the conditions necessary for widespread business disruption.

Gruman’s list includes most of the major problems that arise when managing a large number of desktop or mobile computers. Yet, it is not a complete list. Based on my experience with managing networks, end-user devices, and security, I would like to add the following:

- Spread of malware. With the satisfied need to connect every device to every other device, malware infestations are often able to run rampant across enterprise networks. Further, eliminating malware from end-user devices often means re-imaging or replacing them with freshly imaged systems.
- Data leaks. Traditional fat client PCs can be used to locally store sensitive information. Storage of information might be intentional or incidental to application execution. Intentional data storage results from users making a conscious decision to store information on their local disk. Incidental storage occurs when application information is cached locally to meet processing requirements. In both cases, information is available to theft or loss.
- Controlling access to applications. Once an application is installed on a user’s computer, it can be all but impossible to remove access.
- Providing application/data access to mobile users. The problem here goes beyond access, which can be provided by many solutions, such as SSL VPN. Rather, maintaining mobile device patches and application updates can be difficult.

Most, if not all, of these challenges exist to some degree in any medium to large enterprise. Traditional approaches to dealing with them have been time consuming,

expensive, or fall short of effectiveness objectives. So how did we get ourselves into this mess, and is hope to be found in the next generation of networking technologies?

## The Past, Present, and Future

It's taken us 30 years to arrive at our current state. Moving from centralized processing and management to a distributed model seemed to be a natural path given the nature of PC computing. See Figure 1.

1970s	1980s	1990s	Today
Dumb Terminals Mainframe Processing Collect Keystrokes Green-screen text	Minicomputers Personal Computers Client/Server Fat Clients	Layered Technology Presentation Logic Data Access Data Storage	Virtualization Centralized App Management

**Figure 1: End-user computing timeline**  
(Anthes, 2006)

In the 1970s, the mainframe was king. All processing was performed on a single computer with data retrieved from and stored to storage located in the data center. Access to the data center was tightly controlled. In many cases, users received reports from the computer operators through a window or slot. They accessed electronic information with dumb terminals with no local processing capabilities. The terminals were simple devices that collected keystrokes and presented data in green-screen text.

Distributed processing began in the 1980s, with personal computers finding their way to the desktop. These were fat clients that participated in client/server configurations and connected to the mainframe's smaller cousin, the minicomputer. Although many companies still performed the bulk of their business processing in a centralized processing environment, both applications and data began to drift out to the endpoint devices.

During the 1990s, another shift in business processing architecture took place with the advent of layered system technology. This included building applications with presentation and data access logic layers, with data residing on database servers in the data center. Still, fat client endpoint devices continued to run applications, and more data than ever before found its way to local hard drives. This was also a time when malware writers began perfecting their art. Attacks that eventually spread across entire enterprises often started on an unprotected—or weakly protected—personal computer.

In the 21<sup>st</sup> century, IT managers began to realize that traditional methods of managing desktop and laptop systems were no longer effective in dealing with changes in business requirements, user demands regarding technology implementations, and black hat hackers transitioning from fun and games to an organized crime business model. Demands for the rapid turnaround of application installation or upgrade requests, the need to quickly apply security patches to operating systems and applications, and many

other management headaches are driving a new approach to endpoint processing and management—a combination of desktop virtualization and application streaming.

## Desktop Application Virtualization

For the purpose of this paper, I define desktop virtualization as a state in which application runtime environments, running on the users' desktops, execute within a private space, sharing only operating system services and hardware. For clarification, let's examine what this does not include.

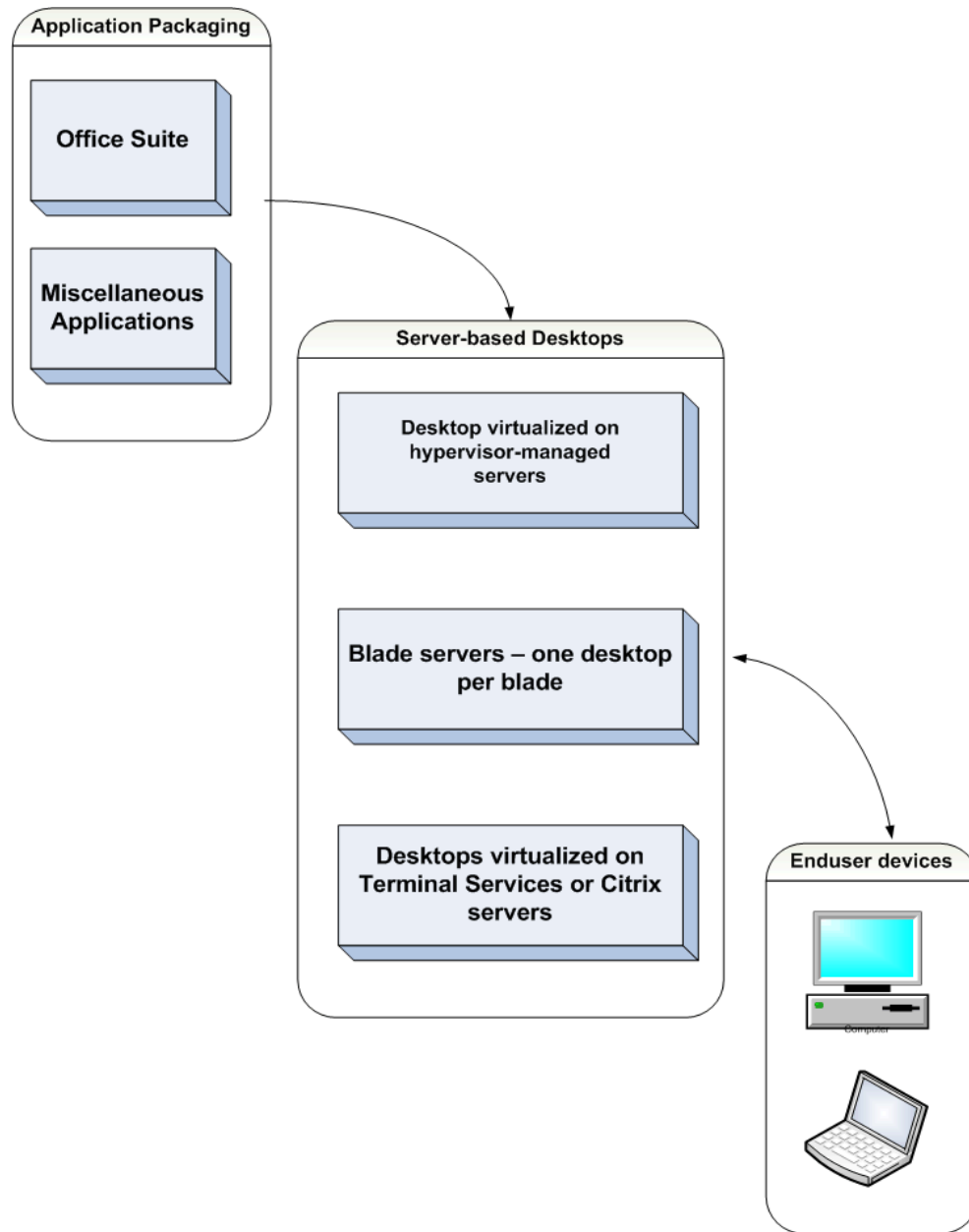
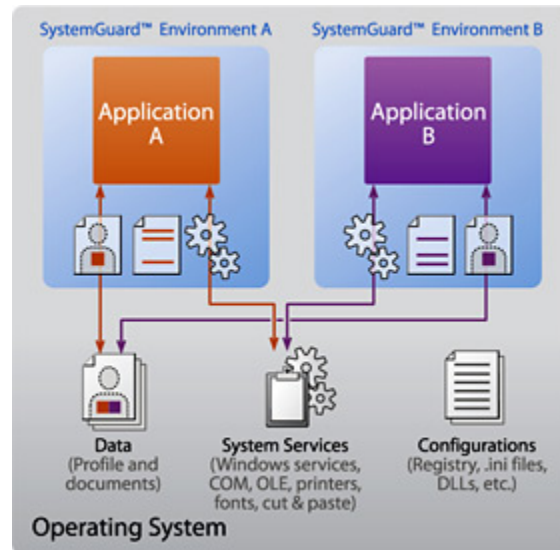


Figure 2: Server-based desktop virtualization

Figure 2 is what many think of when discussing desktop virtualization. Instead of permanently installing applications on users' endpoint device, they are installed in virtualized server environments. They might also be installed on blade servers, with each blade corresponding to a single desktop device. Finally, thin clients accessing terminal services or Citrix sessions are sometimes running virtualized desktops using solutions like SoftGrid. These are examples of server-based computing, and they do not fall within the definition of "virtualized desktops" used in this paper. One big problem with this approach is leaving unused a significant amount of processing potential in the form of desktop fat clients. Now let's take a look at a virtualized desktop environment that uses the power available.

Figure 3 depicts a Microsoft SoftGrid-enabled desktop. Each application runs in an isolated environment. Although the applications share OS services and hardware resources, components unique to each application (e.g. registry entries, dynamic link libraries, COM objects, etc.) can be private to that application—running within the application "sandbox." This approach does not virtualize the OS: just the applications.



**Figure 3: Microsoft SoftGrid virtualized desktop (Microsoft, 2007)**

Isolating application processing environments resolves several of the challenges surrounding endpoint availability, including:

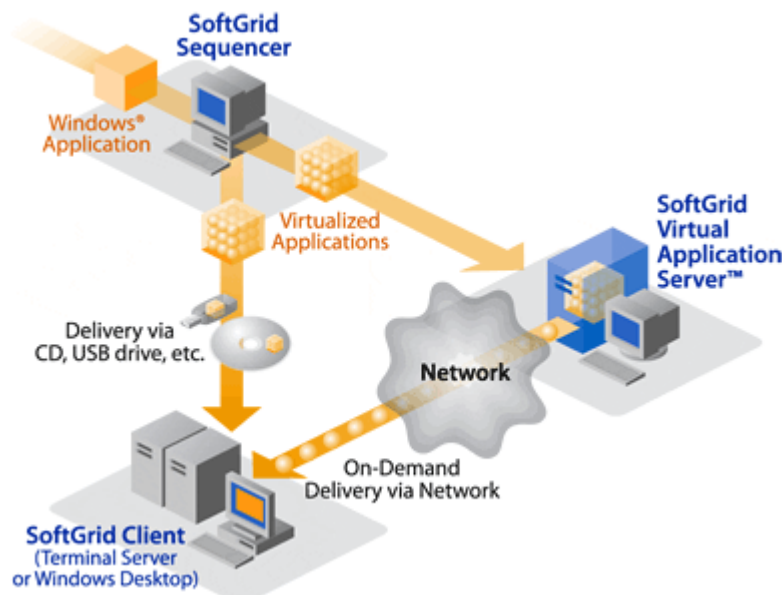
- Incompatibility between application components. Applications deployed and running within a sandbox can use their own private copies of the registry, DLLs, and other components that might conflict with other applications. This also positively affects Help Desk costs.
- Spread of malware. When an application running in virtualized space is infected by malware, the infection is unable to spread to the rest of the PC.
- Data leaks. Data leaks from cache are less probable. In a virtualized application environment, local cache is unique to each application instance running. Shut down the application and the cache is eliminated.

Now we will examine the second step toward a more secure and available desktop environment—application streaming.

## Application Streaming

The second piece of this potential solution for endpoint availability and security management is centralized distribution and management of applications. There are two ways to accomplish this. First, entire applications can be downloaded to virtualized runtime environments. Second, only those components necessary for initial load and execution of the virtualized applications are downloaded. Additional components are downloaded as necessary. In this paper, we will use the second method to explain the benefits of desktop application virtualization.

VMware and Microsoft SoftGrid are two of the biggest vendors for application virtualization. I chose SoftGrid for our walk-through of application virtualization technology.



**Figure 4: SoftGrid Application Packaging**  
(Microsoft, 2007-b)

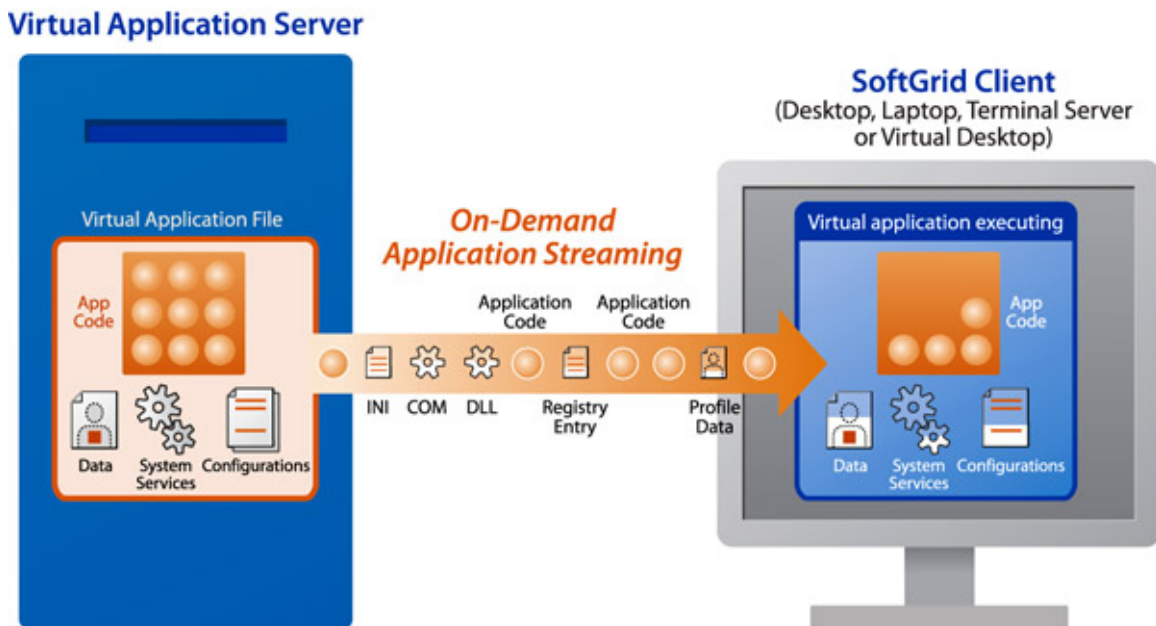
SoftGrid is capable of packaging applications for on-demand, streamed delivery into virtualized endpoint runtime environments. Figure 4 depicts the first step in this process.

A workstation is configured with the SoftGrid Sequencer application. As the SoftGrid administrator installs the target application on the workstation, the sequencer monitors all installation steps, including changes to the registry. The administrator can also select specific components to be included in the virtualized application package, such as DLLs as well as Java and .NET components. Further, the application can be configured to store information in a centralized location (e.g. a secure data center).

The final outcome of the sequencer process is a set of four files that comprise the virtualized application, with an initial application load just big enough to load and initially execute the application. According to Microsoft, the load size is approximately 20 to 40 percent of the total application size (Microsoft, 2007-b).

The four files are placed on a SoftGrid application server for distribution. The administrator grants access to the application by adding users with approved access to a related AD group. Only members of the group will be able to see the application icon on their desktops or access the application files on the server. To reverse the process—to revoke a user’s access—simply remove him or her from the group.

Once a user is added to the proper group, the application icon will appear on her desktop at next login. If the user is already logged in, she can force a refresh of her desktop by using a SoftGrid utility typically found in the system tray. The application is accessed by double-clicking the icon. Figure 5 depicts what occurs when the user runs the application for the first time.



**Figure 5: SoftGrid Application Streaming**  
(Microsoft, 2007)

The four files created and installed on the SoftGrid Application Server are accessed by the desktop. The result is the creation of a virtual application environment on the user’s machine with the bare minimum of application components streamed into it. The result is a self-contained application runtime space that virtualizes the following components (Microsoft, 2007-c):

- Registry – registry changes unique to the application are not made to the main OS on the desktop. Rather, they are virtualized within the isolated application runtime space.

- File system – calls from the application for local disk access can be redirected to access DLLs and other components from a virtual file system.
- COM/IPC
- INI files
- Process environment
- Fonts

Desktop users can access the virtual applications for which they have access from any workstation in the enterprise or from home. Laptop users can pre-stage the application for offline use.

If an administrator wants to upgrade an application, she simply installs it on the sequencer, building four new SoftGrid files. If it's necessary to run the new version in parallel with the old, the new application package is stored in a new folder on the SoftGrid server. Users granted access to the new application can run both the old and the new versions at the same time on the same desktop, since they are running in virtualized environments.

### **Meeting Management's Challenges**

The management challenges listed earlier this paper are addressed by streaming applications to virtualized application processing environments in the following ways:

- Applying patches in a virtualized environment is a simple rebuild of the appropriate SoftGrid package. The next time a user runs the application, the updated version is automatically streamed to the desktop.
- Help Desk costs associated with failed application installations, overwritten application components, corrupted registries, etc. are all but eliminated when files and settings unique to an application are virtualized.
- Use of applications accessed via the SoftGrid server is tracked. Further, administrators can link an active instance of a running application to a license. This metering of applications helps organizations remain compliant with licensing agreements.
- If a virtualized application environment is infected with malware, the threat is contained—prevented from spreading to other applications or the base operating system.
- The threat of data leaks is mitigated due to the virtualization of local cache associated with application processing. Further, configuration of applications such as the Microsoft Office Suite can 'encourage' users to save documents in a secure, centralized environment.
- Application access is controlled by group membership. In addition, applications that run on laptops can be configured to stop running if the user doesn't

authenticate to the enterprise network within a specified period. This prevents thieves from using laptop applications indefinitely.

- Mobile users are able to access patched or updated applications from any location.

## **Conclusion**

The proliferation of fat clients and distributed applications has been the result of a natural progression from centralized, inflexible computing to decentralized computing focused on user productivity. Managing this new world has been difficult for IT and Security teams. This, coupled with the rising cost of licensing and Help Desk operations, is causing organizations to look for a 'better way.'

Although application virtualization is not the only solution to today's desktop management challenges, it is positioned as a viable alternative.

---

© 2007 Thomas W. Olzak.

Tom Olzak, MBA, CISSP, MCSE, is President and CEO of Erudio Security, LLC. He can be reached at [tom.olzak@erudiosecurity.com](mailto:tom.olzak@erudiosecurity.com).

Check out Tom's book, [Just Enough Security](#)

Additional security management resources are available at <http://adventuresinsecurity.com>

Free security training available at <http://adventuresinsecurity.com/SCourses>

---

## Works Cited

- Anthes, G. (2007, March). "The Virtual Desktop is here", *ComputerWorld*, retrieved June 29, 2007 from <http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=282928>
- Gruman, G. (2006, September). "Desktop virtualization: Making PCs Manageable", *ComputerWorld*, 12 Sep 2006. Retrieved June 29, 2007 from <http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9003227>
- Microsoft (2007). *Application Virtualization*. Retrieved 13 Aug 2007, from <http://www.microsoft.com/systemcenter/softgrid/evaluation/virtualization.mspx>
- Microsoft (2007-b). *Microsoft SoftGrid application virtualization for Systems Management Server*. Retrieved 14 Aug 2007 from <http://www.microsoft.com/systemcenter/softgrid/evaluation/delivery.mspx>