Realtime publishers

VDI or Desktop Virtualization: What's Right for You?

The Essentials Series

sponsored by

WANOVA®

Greg Shields

Introduction to Realtime Publishers

by Don Jones, Series Editor

For several years now, Realtime has produced dozens and dozens of high-quality books that just happen to be delivered in electronic format—at no cost to you, the reader. We've made this unique publishing model work through the generous support and cooperation of our sponsors, who agree to bear each book's production expenses for the benefit of our readers.

Although we've always offered our publications to you for free, don't think for a moment that quality is anything less than our top priority. My job is to make sure that our books are as good as—and in most cases better than—any printed book that would cost you $40 or more. Our electronic publishing model offers several advantages over printed books: You receive chapters literally as fast as our authors produce them (hence the "realtime" aspect of our model), and we can update chapters to reflect the latest changes in technology.

I want to point out that our books are by no means paid advertisements or white papers. We're an independent publishing company, and an important aspect of my job is to make sure that our authors are free to voice their expertise and opinions without reservation or restriction. We maintain complete editorial control of our publications, and I'm proud that we've produced so many quality books over the past years.

I want to extend an invitation to visit us at http://nexus.realtimepublishers.com, especially if you've received this publication from a friend or colleague. We have a wide variety of additional books on a range of topics, and you're sure to find something that's of interest to you—and it won't cost you a thing. We hope you'll continue to come to Realtime for your educational needs far into the future.

Until then, enjoy.

Don Jones

Copyright Statement

© 2011 Realtime Publishers. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtime Publishers (the "Materials") and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtime Publishers its web site sponsors. In no event shall Realtime Publishers or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Realtime Publishers and the Realtime Publishers logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

If you have any questions about these terms, or if you would like information about licensing materials from Realtime Publishers, please contact us via e-mail at info@realtimepublishers.com.

Article 1: Aren't VDI and Desktop Virtualization the Same?

Often, when the decibel level of a conventional wisdom reaches a feverish pitch, it can be wise to seek the opposite of what the convention suggests. Examples of this wisdom happen all the time in our daily lives. When the masses refinanced into 3- or 5-year ARMs, the smart ones held their 30-year fixed mortgages. Investors follow similar behaviors. The irrationally exuberant moved conservative-but-reliable securities into high-risk tech stocks and learned how fast the bubble bursts when "everyone else is doing it too."

In the past few years, those same behaviors have been infiltrating our industry. For us, the topic isn't mortgages or investments. Rather, it's the approaches we use in delivering desktops to users.

"VDI is the answer, and the answer is VDI!" is a common theme you'll hear as you attend IT conferences or read industry magazines. Yet the exuberance that surrounds VDI technology belies the fact that VDI is but one approach to desktop delivery. VDI is also only one of many options to benefit from desktop virtualization. Unfortunately, as too many find out far too late, VDI might not necessarily be the solution to your users' needs.

There's a Difference?

If this series' title confuses you, don't fret. You're not alone. Thanks in part to an impressive volume from marketers and aficionados alike, VDI has quickly elevated from niche topic to "something you absolutely must do." Although VDI technology is undeniably exciting, VDI and desktop virtualization aren't necessarily the same thing. In reality, *VDI is but one of many ways in which desktops can be virtualized*.

VDI represents an approach to desktop delivery, just like manually installing a user's desktop via DVD media. VDI's primary difference stems from its centralization. VDI deployments centralize desktops into the data center, enabling IT to act as a service provider. From this location, IT gains operational efficiencies over the traditional physical desktop delivery approach, such as the ability to manage just a single copy of Windows, having all others operate as clones from that template original.

Yet many who are considering VDI don't realize that it presents IT gains as well as losses. Users tend to lose in the deal as well. VDI's approach functions exceptionally well for task workers who consume static application sets. It fits lab environments that require rapid deployment and turnaround. It can also be a perfect solution for users with light application sets who never leave the confines of their local high-speed LAN. VDI's undeniable charisma, however, reveals its true limitations the moment those workers move outside the brick-and-mortar. Attempt to push Adobe Flash, video, or voice over WANs, and users quickly discover the use cases where VDI fails to meet their needs.

Highlighting this contrast is the goal of this series. In it, we'll explore what desktop virtualization really aims to accomplish. You'll also learn about virtualization approaches that may provide a better fit than VDI for your situation. Focusing first on the needs of the user and second on the improvements offered by virtualization to IT automation, you'll see the big failures traditional VDI introduces to the social contract between users and IT. On this journey, you'll come to understand how one particular approach—*hybrid desktop virtualization*—combines the best of the others to merge IT's centralization desires with users' real-world requirements.

Finding Balance Between Centralization and Flexibility

The guiding notion behind desktop virtualization is IT's desire to optimize asset management. Look back not that far in the past and you'll find all kinds of terrifically-ineffective examples of our early management practices:

- Walking around with DVDs to install software
- Possessing zero configuration control and even less update compliance awareness
- Employing manual processes in applying operating systems (OSs) to desktops
- Boasting no or limited ability to transfer user state between devices—laptop to desktop, desktop to conference room PC, others—while synchronizing changes among all devices

Although automation toolsets indeed brought some centralized control to these activities, implementing the toolsets began an industry obsession with strategies that favor control above all else. At fault is the very real human nature to seek solutions that make one's life easier. Nearly any IT administrator will tell you, "Everything would be much easier if we could better lock down our desktops."

Yet although control and lock down absolutely fit IT's goal of making their job easier, this heavy-handed approach hinders the users' experience. Ever worked in a fanatically-locked-down environment where every personalization element is defined for you, every application is nailed down tight, and every data access is check pointed by multiple security controls? The experience isn't pleasant. If this environment embodies your administrative mindset, take off your Domain Admin gloves some day and try working as a regular user for a while. You won't like it either.

That's why the emerging IT conventional wisdom recognizes that a single-minded control focus will never achieve desired goals. The reality is that *users will always find a way around*. Increase password complexity, and users will write them down on sticky notes. Take away their personalization, and they'll construct elaborate business justifications why personalization must remain. Move their desktop into the data center, and they'll just switch to their (far less trustworthy) home laptop to finish their work.

All this said, centralization is important for management, security, and simply keeping people connected in appropriate ways. For the right desktop virtualization for your environment, you need to find the best balance of centralized management and control while maintaining the best possible user experience and mobility.

So What Is Desktop Virtualization?

If VDI is only one approach to desktop virtualization, what really is this superset? What are the different ways in which desktops can be virtualized, and why? The most commonly-known approaches typically arrive in one of two forms: *server-based desktop virtualization* or *client-based desktop virtualization*.

Server-based desktop virtualization represents the traditional VDI concept. With it, desktop processing executes inside the data center in a pool of specially-created virtual machines. Users access those virtual machines using one of many remote protocols (ICA, RDP, PCoIP, and so on). Most solutions leverage pools of desktop clones that are provisioned to users at the point of login. Applications, configurations, and personality are delivered on-demand during the provisioning process.

Most server-based desktop virtualization solutions are forced into this pooled architecture due to the high cost of storage and IT's desire for centralized control. Keeping things running smoothly requires limiting personality elements and the set of applications delivered to the provisioned desktop. With shared resources, a user running heavyweight applications won't have a good experience when inappropriately collocated with others on a virtual host. Server-based desktop virtualization also has the oft-overlooked effect of relocating large quantities of data away from inexpensive desktop storage and onto expensive SAN storage. *It's the same data; you just need more dollars to keep it around.*

Client-based desktop virtualization relocates the desktop's processing back to the desktop. The user still interfaces with a virtual machine, but that virtual machine is stored locally on the user's physical computer. There, the local processors, memory, storage, and networking are used in processing the virtual machine's activities.

The biggest user experience benefit of client-based desktop virtualization is in eliminating the reliance on a remote protocol. The computer resides locally, so the OS instance can follow the user as they move around. Taking a laptop outside the confines of the LAN means still having access to applications and (locally-copied) data without the need to first find a fast-enough network connection. It also means being able to horizontally scale the environment without resorting to comparatively-expensive server-class equipment. With client-based desktop virtualization, scaling out means buying a few more desktops or laptops. That's great for the pocketbook.

Yet this approach isn't without intrinsic problems as well. Most client virtualization requires some form of hypervisor in order to do its job. These hypervisors tend to come in one of two types: installed directly to the computer's hardware or installed into another OS.

Whereas the hypervisor has been such a boon to optimizing servers, using it at the desktop doesn't bring an immediate win. In the first hypervisor case, the directly-installed (Type-1) hypervisor must be aware of, or able to become aware of, every single device ever connected to or inside a computer. That's *millions* of drivers, which represents a project so enormous that any of today's directly-installed hypervisor solutions suffer under the weight of a severely limited hardware compatibility list.

The second case isn't much better. Installed on top of an existing OS, the second (Type-2) hypervisor architecture eliminates the driver problem but has the effect of doubling your number of systems under management. The first is the host OS, and the second is the virtual machine. That situation is not more efficient, it's less so.

Finding a Hybrid

As you can see, desktop virtualization at both the server and client side arrive with their own set of detractors. The server-side case improves centralization but at the cost of flexibility and mobility. The client-side case reverses that equation. Arguably, with only these two options at one's disposal, it is easy to see why VDI remains a viable solution: *the cons of client-side option don't outweigh the pros of server-side setup.*

But what if some mechanism existed that could hybridize the best parts of both approaches while eliminating their cons. Such a solution might actually find that perfect balance between IT's centralization goals and users' need for flexibility. That solution can exist, if architected with the previously mentioned pro/con list as its requirement specification. One possibility is hybrid desktop virtualization, an explanation of which you'll find in the third article.

But before we get there, we need to further nail down the exact reasons our existing VDI technologies fail us. That understanding will help you realize why something fundamentally different is in fact superior. The next article in this series provides a review of VDI's big failures.

Article 2: A Look at Traditional VDI's Five Big Failures

For too long, VDI has been seen as *the next era in desktop delivery*. This attribution implicitly presumes that VDI is to be the panacea for how all desktops are to be provisioned for their users.

Today's thinking is slowly shifting from that belief. First and foremost is the recognition that different use cases have very different requirements, not all of which are solved by virtualizing your desktops and throwing them into the data center. Although some businesses indeed jumped on VDI's bandwagon as early adopters, signs appear to indicate its rate of adoption is slowing. Qualifying that statement, smart businesses recognize that VDI's adoption makes sense for the special cases where it best fits—rather than being everything for everyone.

Rare is the company that has moved its entire employee base off physical computers and onto VDI. One reason for that stagnation could relate to the mismatch between the needs of IT and its users, which the previous article explored. Why do many VDI projects still remain in the pilot phase? A potential explanation is that implementing it to the wrong use cases introduces a set of big failures. What are those failures?

Square Peg. Round Hole.

The irrationally exuberant might loudly proclaim that, "We'll be moving all of our users to VDI in this project!" Yet that same individual might not realize that inside most companies exists a range of user classes. Consider the following example classes that might exist inside your environment:

- **The task worker.** These individuals typically interact with a very limited set of applications. Task workers by definition are paid to accomplish a predetermined set of tasks using a closed set of tools. The daily workflow of task workers typically sees them working with a desktop computer inside the LAN and rarely continuing that work outside. Some task worker situations also use hot desking, where workers do not occupy the same physical space during each work shift.
- **The knowledge worker.** Knowledge workers are typically not tasked so much as handed rough goals and deadlines. Their managerial emphasis leans toward independence and self-determination. As a result, knowledge workers tend to require a larger and more dynamic suite of applications to complete their work. Their greater level of professional freedom also tends to impact their desire for a more customized workspace.

- **The software developer.** Software developers are a special class of knowledge workers in that their applications tend to require low-level administrative access with high resource consumption. Their application set is also highly dynamic and may include applications that don't integrate well into automation solutions.
- **The never-in-the-office worker.** Notwithstanding this worker's class of user, the never-in-the-office worker presents a special case. Their hardware rarely if ever returns to the LAN, which often means they require administrative rights to self-service problems and add applications. Their computers lie outside the LAN's administrative boundary, so they often represent a higher threat level.
- **The sometimes-outside-the-office worker.** Much different than the "never" case is the sometimes-outside-the-office worker. Executives, salespeople, consultants, and the like, these people spend some portion of their job inside the LAN and another portion outside. This group also has special needs due to the constantly varying trust level associated with their hardware as well as their changing access requirements.
- **The outside consultant.** Not an employee, the outside consultant typically enjoys far less freedom inside the office. Outside consultants are typically brought in to accomplish specific tasks, with accesses being highly controlled to prevent exposure. These individuals are only partially trusted, and must be delegated responsibility, applications, and accesses with care.
- **The worker in the conference room.** Every person inside the office occasionally needs to leave their workspace to join others in a conference room. While there, application sets are rarely dynamic; however, maintaining the user's workspace in order to facilitate the meeting is always a desire.
- **The lab trainee.** Finally are individuals in a lab environment, whether for testing or learning. Lab environments represent special cases as well, with rapid rebuild and rapid provisioning being key goals for efficiency.

It should be plainly obvious that these different sets of individuals are best serviced through very different desktop delivery mechanisms. Although task workers, outside consultants, conference room workers, and labs might all present a perfect fit for VDI's desktop provisioning model, the absoluteness of VDI's efficacy grows hazy as one analyzes the other use cases.

Is it better to give the never-in-the-office worker a VDI desktop or allow them to work with their uncontrolled home computer? Will a VDI desktop help the sometimes-in-the-office worker while they're at their desk but significantly impede their activities once they leave? Can the advanced requirements of knowledge workers and software developers be met by VDI's everything-for-everyone delivery model?

Traditional VDI's Five Big Failures

If your answers to those questions have you concerned, know that you're on the right track. VDI's entire delivery approach is well-suited to a particular set of use cases. Individuals with LAN-speed connections, limited need for customization, relatively static application sets, low resource requirements, and a rare requirement for access outside the office all represent round pegs for VDI's round hole.

However, there are use cases where even today's VDI technology advances can still lead to failure. The following list highlights the biggest five to watch out for.

Failure #1: Latent Network Connectivity

Protocols like RDP and ICA are designed to be exceptionally bandwidth tolerant. One can run RDP or ICA through extremely narrow network connections and expect an acceptable user experience.

These protocols, however, also tend to be latency intolerant. The reason relates to the types of activities being done in the session. Clicking a mouse, entering text into a document, or moving a window works very well when latency is effectively zero. That same experience becomes very dissociative when latency grows above 200 or 300 milliseconds. Even with caching technologies, that extra fifth or third of a second between action and reaction usually makes for an unacceptable user experience.

Failure #2: Heavy Applications

VDI gains its cost efficiencies by collocating many virtual machines atop a smaller number of hosts. It rewards smart administration when more users work atop less hardware. Yet those benefits quickly unravel once "heavy" applications are provisioned to desktops.

These heavy applications—Adobe Flash et al, multimedia, CAD/CAM, imaging, and so on—consume comparatively larger quantities of system resources than do their lighter brethren. More still equals more, even considering today's advancements in resource optimization. Thus, supporting heavy applications incurs a comparatively greater cost per provisioned desktop than in the lightweight model.

Failure #3: Video Conferencing, Voice, and Softphones

Communication technologies that bring people together for live conversations require low latency if those conversations will have value. If you've ever placed a call over a satellite phone or poor VoIP connection and heard the multi-second delay between speaking and hearing, you recognize the special challenge network latency creates for these applications.

At issue is not necessarily the technologies that enable these communication mediums to work within VDI. Vendors today are making great strides in caching and other technologies that limit the impact. A much more operational issue is the dynamics of the environment itself. Although 20 virtual desktops on the same host might see their communication platforms perform flawlessly, that same experience can quickly degrade when hardware resources go into contention.

Failure #4: Highly-Dynamic Application Sets

Vendors today have also created mature technologies for rapidly deploying applications to VDI desktops. The process is little different than with physical computers. The sometimes unrecognized hurdle is that every application automation solution first requires prepackaged applications. That packaging process takes time and costs money.

The cost/benefit analysis gets worse when applications are not commonly used. Consider the one-off situation where a single (perhaps homegrown) application is needed by one or a few users. Here, the effort to package can far outweigh just simply installing it via Next, Next, Finish.

Automation grows even less effective when such applications require regular updates. Manual installations don't fit into the VDI approach due to the fact that logged out desktops are typically cleansed and returned to an available pool. In essence, you would need to continually Next, Next, Finish any needed application at every logon. Not good. Creating special cases for one-off users and their applications impacts VDI's cost model as well as its administrative optimizations.

Failure #5: Offline Use

Not every user in a business accomplishes their tasks as they sit at their desks or are attached to a high-speed network connection. Some work in client or partner sites where connectivity isn't permitted. Others travel to places where connectivity is unavailable at worst and spotty at best.

Traditional VDI's solution to offline use often involves a check-in/check-out process whereby a virtual machine is transferred from the data center to the user's laptop. The sheer size of most virtual machines makes this a lengthy process. How much time did your last 40-plus-gigabyte file transfer require? More time than patience, one supposes.

Seeking a Middle Ground

Like so much in life, it seems that a focus on either of the opposing options outlined thus far suggests the real solution lies somewhere in the middle. Whatever product filled that gap would enjoy all the centralization functionality gained by storing desktop images in the data center as well as the flexibility of processing those images on local hardware. The final article in this series discusses how one approach, hybrid desktop virtualization, helps align these two needs under a potential single solution.

Article 3: Hybrid Desktop Virtualization Aligns IT's Needs with End User Requirements

This series has perhaps unfairly pointed a finger at VDI for being insufficient across many business use cases. The reality is that both server-based and client-based desktop virtualization have their detractors. In the server-based (VDI) model, consolidating desktops into the data center moves the processing off perfectly good hardware and onto perfectly new servers. Great idea, bad execution. In a straight comparison, a server is always more expensive than a desktop. So unless you have those servers lying around, you're in for a capital expense.

What many potential VDI adopters don't often realize is that even with relocating that processing, *there still must be some device at the endpoint to receive the virtual machine*. If that machine is the same old desktop, you're now running double-desktops. We've already agreed that's a bad idea. However, if you switch to thin clients, you'll find yourself with yet another capital expenditure.

The client-based approach isn't that much better. In a world where a Type-1 hypervisor was available that supported every possible hardware configuration, the client-based approach would absolutely be an attractive option. But that world currently doesn't exist. Type-2 solutions are available, yet they create the same unexciting double-desktops situation, and have been notoriously poor in terms of overall performance.

The everything-for-everyone end state offered by conventional wisdom's VDI decibel level begins to get muddied the more you analyze desktop virtualization's potential users and use cases. Dig deeper into your own classes of users and you'll find that the best solution is probably a combination of both.

Yet what still lacks in all the options discussed so far is that priority on preserving the users' experience. Users demand the ability to customize their workspace. They also, often without realizing it, demand local application processing in those circumstances discussed in the previous article. At the same time, IT must implement centralization to succeed in managing a growing environment that only gets more complex over time. Merging these two requirements is the goal of desktop virtualization in general.

Hybrid Desktop Virtualization and "Layering the OS"

Seek solutions in the middle. The correct one leverages all the centralization tactics proffered by VDI while retaining the local execution users demand. One way to get there can happen through a focus on *desktop synchronization*, or what is also called hybrid desktop virtualization. In this architecture, it becomes possible to achieve the goals of IT while giving users what they want.

The hybrid model abandons the server-based model's movement of desktop processing into the data center. It at the same time eschews the notion of delivering a very large virtual machine to local hardware only to exist atop something else.

In place of these two opposing options, the hybrid approach works with the operating system (OS), applications, and data that are installed directly to the user's computer. Beginning with an OS template, the hybrid model deconstructs the monolithic notion of "the user's computer" into a series of stacked layers each of which can then be individually managed.

Although not specifically focused on the hybrid model's deconstruction, I introduce the notion of OS layering in a 2009 *TechNet Magazine* column called *A Case for a Layered Approach to Deploying Windows Desktops* (http://technet.microsoft.com/en-us/magazine/ee835710.aspx). In that article, I suggest that,

> …your average Windows desktop is a lot like the layers of an onion. At its core is an operating system with default and out-of-the-box settings. Installed atop that core are the necessary drivers as well as required OS updates and patches. Next up are the user's needed applications, along with individual configuration settings that define your custom desktop environment and modify application settings. Finally, the top layer includes the user's specific personality data such as bookmarks, desktop shortcuts and printers.
>
> All of these individual layers combine to create what ultimately becomes the user's working environment. If you peel back each layer, you eventually make your way back to the core operating system itself.

Later in the article, I reformat the text in the quote above into a graphical format. That image I've reprinted here (see Figure 1) to visually separate each layer in relation to those that sit above and below. Although this image (and the article) reflects Microsoft's free tools that manage the content within each layer, you can obviously imagine the range of third-party solutions that handle the behaviors of each.

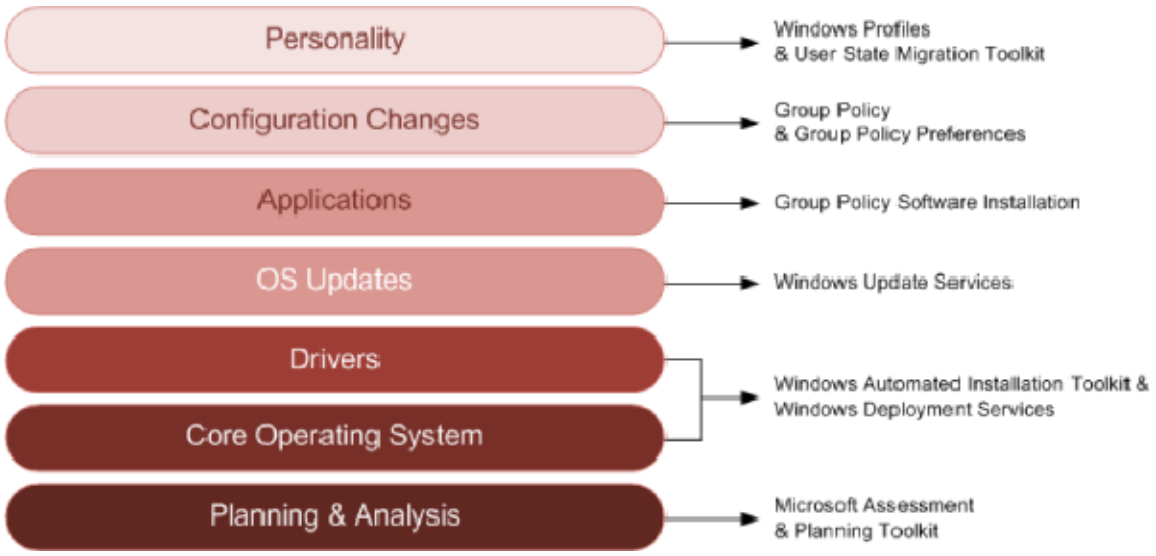

Figure 1: Layers of the Windows desktop.

Significantly different between my original layering concept and the architecture that is hybrid desktop virtualization is the addition of a synchronization function. Imagine, if you will, that your desktop has been deployed using a solution that follows the hybrid model. In doing so, a base image has been laid down on disk, followed by a series of applications and specialized configurations. After that come your user personality elements.

Different than the typical Windows installation, this deployment is provisioned via synchronization from a centralized server and template image. Once deployed, your subsequent changes to that image are themselves synchronized back to the central database. Anything you do to impact one of the layers depicted in Figure 1 is automatically and transparently replicated somewhere else.

The operational result of this bidirectional synchronization completely changes IT's actions in managing its computers. Your personal desktop synchronizes to a server so that losing it or experiencing a non-recoverable problem is resolved by merely resyncing a previous version back to your hardware. Lose the hardware, and you need only resync the instance to something different.

Layered applications and personality follow you around as well. Leave your desktop to enter a conference room and you can fully expect your personal settings to follow you into that location. Shift from a desktop to a laptop or vice-versa, and the same scenario applies.

IT gains incredible efficiencies as the synchronization goes in the other direction. When IT needs to deploy an update, IT need do so only to the OS template on the server. Applications work in much the same way. They can be either fanned out in an automated fashion or directly installed via the manual method. Each use case benefits.

Seeking the middle needn't necessarily require abandoning the options at either end. The hybrid approach's greatest potential lies in its ability to pair with existing server-based desktop virtualization solutions. Just like VDI, the hybrid approach won't necessarily work well across every use case. That said, its layering does present the possibility where server-based processing can be accomplished when server-class equipment (or data center-grade security) is required. Synchronizing that instance down to local hardware needn't require long delays because much of the OS instance is mostly already cached locally.

Hybrid Desktop Virtualization Bridges VDI and Client

VDI may not be the answer, and the answer may not be VDI. But its technologies are important for those special cases. In others, the client approach works well. For those in the middle, hybrid desktop virtualization bridges the gaps in VDI's coverage. At the end of the day, your goal must be in maintaining user experience while ensuring the safety and security of the environment. Getting there sometimes requires taking the middle road.