<table>
<thead>
<tr>
<th>1</th>
<th>Desktops today</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Desktop Virtualization</td>
</tr>
<tr>
<td>3</td>
<td>How to talk to the Server Virt Guy</td>
</tr>
<tr>
<td>4</td>
<td>Why do people like VDI?</td>
</tr>
<tr>
<td>5</td>
<td>Why VDI?</td>
</tr>
<tr>
<td>6</td>
<td>I think people are confused on the difference between “TCO” and “ROI”</td>
</tr>
<tr>
<td>7</td>
<td>Windows 7</td>
</tr>
<tr>
<td>8</td>
<td>Devices</td>
</tr>
<tr>
<td>9</td>
<td>Screw Client Hypervisors</td>
</tr>
<tr>
<td>10</td>
<td>App Virtualization for everyone</td>
</tr>
<tr>
<td>11</td>
<td>Desktop Virtualization and the Network</td>
</tr>
<tr>
<td>12</td>
<td>Desktop Virtualization storage</td>
</tr>
<tr>
<td>13</td>
<td>AD &amp; SIDs</td>
</tr>
<tr>
<td>14</td>
<td>How VDI affects security</td>
</tr>
<tr>
<td>15</td>
<td>The Windowless story</td>
</tr>
</tbody>
</table>
The state of today's desktop

Ah... the desktop. We love you!

Well, most of us say "desktop" even though we use laptops.

But the desktop has been around for what...20 years?
Actually, heck, it's been almost 30 years.
Yikes!
And I guess when it comes to a work desktop, we're pretty entrenched in the way things work, which is something like this:

1. IT builds a desktop/laptop for a user
2. User starts using it
3. User breaks something
4. Repeat ad infinitum

So actually... hmmm... this kind of sucks.

And it sucks from everyone's perspective, both from the IT point of view and the end user point of view.
Benefits for IT

There are a lot of reasons that IT folks want to virtualize users' desktops, including:
- Improved control
- User's aren't tied to specific devices (easier support)
- Easier backups
- Faster installs / moves / adds / changes

Benefits for the business

- Increased productivity (faster app provisioning, etc.)
- Ensure compliance (for security and regulatory reasons)
- Increase security

Benefits for end users

- Freedom to work where and how they want
- Freedom to choose whatever device they want
- Don’t have to worry about backups & compliance
- Easier separation of “work” and “personal”
What is desktop virtualization?

Virtualization is about the separation of the physical from the logical. So taking that definition and applying it to desktops, desktop virtualization is the separation of the management of the desktop from the physical desktop. And of course there are many ways this can work. Desktop Virtualization is more than VDI.

A lot of people focus too much on the "how" of desktop virtualization without thinking about the "why." But people need to understand why, which I guess ties back to the benefits of desktop virtualization in the first place. (Although to get it, you really need to look at the benefits from a few different perspectives.)
Yo - Can you believe this guy?

Forwarded message attached:

From: Stanley James <sjames@redbarnantiques.net>
To: Gabe Knuth <gabe@brianmadden.com>
Subject: Something you said was wrong

Hi Gabe,

Hope you had a good Labor Day weekend! Mine was spent converting all of our desktops on to the hosts in our vSphere environment so that on Tuesday when everyone signs in, they’ll be moved to VDI. We’ve been doing server virtualization for years, so doing desktop virtualization is a no-brainer! We’ll be able to get rid of most of our 20-person desktop support team while still supporting all 2000 users.

I wanted to let you know this because I read something you once wrote about how server virt shouldn’t be considered the same as desktop virt, but I’m here to tell you it works! Tomorrow is Tuesday and everyone comes back from vacation then...I can’t believe we didn’t do this sooner!

Regards,
Stan

This guy, obviously, is going to have some problems, and the main reason for them isn’t technical or political - it’s because Stan Stan the Server Man didn’t talk to the Desktop team.
What happens more often than anyone would like to admit is that Desktop Virt comes from one side of the IT Department or the other, and when that happens it stays there. It’s SO. FREAKING. IMPORTANT. to remember that desktops are fundamentally different than servers in almost every way, so no matter where the directive comes from, the server teams and desktop teams need to become friends. Fast.

The best relationships are built on respect. They’re the ones where the server team and the desktop team each accept that they know their respective platforms best. If you’re in the unfortunate position of one department marginalizing the other, I feel bad for you. Gabe just lost some more hair for you.

Here’s the thing - if you want your desktop virt project to work, you have to get the teams together. Go out for pizza, or have a golf outing. Tell the server guy that you understand that he knows virtualization, and that his knowledge will be required to pull this project off, but that desktops have different requirements than servers.
First, servers are virtualized for different reasons than desktops. Servers are virtualized for reasons like consolidation, efficiency, flexibility, and disaster recovery. Desktops are virtualized to remove Windows from the cubicle and keep everything about their desktop in the controllable confines of the datacenter. Sometimes it’s done for power savings or disaster recovery (ok, I’ll give you one common thing), and sometimes it’s so your boss can run Windows on his iPad at the golf course. The bottom line is that the reasons for adopting desktop virt are different than those for server virt.

"But the technology is the same," they might say. Agree with them. The fact that a hypervisor is running on a single piece of hardware in order to run many isolated operating systems is the same, but the similarities end there.

Tell them how the reality is that server workloads are PREDICTABLE and BORING (don’t use those words). Each server does the same thing, day in and day out. It might have usage peaks and lulls on CPU, storage, and memory, but it’s all very predictable. Desktops are the polar opposite. They’re CrAzY, unpredictable resource consumers that hog CPU and memory while creating a burden on storage as if they got paid by the IOPS. Why are they so different? Users.
Servers don't have users...well...not users like desktops do.
Nobody is logging in to a server and running Outlook, 8 tabs of IE, each on a different Flash-based or social networking site, custom backgrounds and mouse cursors, and a few line of business applications. Desktop users do that all the time!

Desktops have weird requirements that servers don't have, too. Like webcams and iPads, or 3D graphics, media streaming, and bi-directional audio. They have network requirements that go above and beyond those of servers, too, because users are connecting to these things all day, every day via remoting protocols that only add the network requirements of the VM host. All of these things are unique to desktop virtualization that server virt admins have never had to consider.

Server virtualization admins also don't have to worry about everyone coming in at 8:00 in the morning and accessing 2000 personal virtual machines, all stored somewhere other than where they're actually executing (which means there's gonna be data transfer somewhere). They don't
have to worry as much about the network, storage, and CPU all being hit at the same exact time every day. Poor Stan in the email is going to have a rude awakening on Tuesday morning!

Server guys know server things. They know change control, backups, domain architecture, file and print servers, updates, network things—all important things to know. Desktop people, on the other hand, work with logon scripts, imaging, mass deployment, and application installation, troubleshooting, conflicts, and testing. The most important thing a desktop admin knows, though, might just be the crazy-ass users.

The skill set is different, just as the hardware and software requirements for deploying physical desktops are different from those of physical servers. Keeping the users in mind, and the different workloads that they bring to the table is so critical to the success of a project that you can’t just up-and-decide to deploy VDI to your existing virtual infrastructure. You need to plan it, benchmark it, and use real world examples from YOUR environment to establish a baseline.

Call it job security.
The key to everyone getting along is to realize that A) each team knows something important, and B) desktops are different than servers. Even after you figure that out, you can feign an argument here and there to get another golf outing, but the work you'll get done when not golfing will be awesome.

**Why do people dig VDI?**

CIOs

Compliance

Control

Security

Faster patching

End Users

Broken hardware doesn't mean a lot of time away.

Can work at home without carting a laptop back-and-forth.

Don’t have to worry about backups.

IT Guys

Push out patches faster.

Desktops are in the datacenter, so easier to support.
Why VDI?

ugg... there is SO MUCH CRAP out there around VDI. I wonder if anyone's ever made a list of the real reasons that real customers use VDI today? Sooo...Why do people use VDI today?

- They want the benefits of centralized desktops, but they're afraid of Terminal Server
- Virtualization is already cool in their company, and they think that desktop virtualization is just another form of server virtualization (wrong but definitely a reason)
- Flexibility for disaster recovery or pandemics

Why do people NOT use VDI today?
- Perception that remote computing doesn't work?
- Perception that VDI is too complex or not ready yet?
- Perception that VDI is too expensive or that it doesn't actually save money?
- They use terminal server instead?
Come together
Hello people?? To be successful with VDI, you need to get your desktop and server teams talking to each other. Desktop needs to know about stuff like IOPS, hypervisors, and real storage. Server peeps need to know about profiles, app conflicts, and wackiness of users.

Want to get fired? Do these 5 things!
1) Take away what you perceive as the "non essential" things users do.
2) Make it take longer to connect or start up each morning
3) Don’t let them work offline
4) Don’t let them use the devices they want
5) Buy the cheapest thin clients you can

Which kind of desktop virtualizer are you?
A "desktop" person, who approaches desktop virtualization from the desktop "in" to the datacenter.

A "virtualization" person, who approaches desktop virtualization from the datacenter "out" to the desktops.

I’ll bet this affects how you think about your virtual desktop design? (datacenter-centric versus user-centric!)
We all have to get there by 2014, but it’s so new and different. Lots of issues with app compat, new security, new hardware, new profiles, etc. So I’ll talk about how the various desktop virt technologies can help you get there, including app virt and user virt, but also client-based Vms, VDI, etc.

Damn you Windows 7

We all have to get there by April 8, 2014, or we lose our jobs.

<<<<maybe some kind of "Days until I have to get to Win7" with countdown numbers crossed out, like 1233, 1232, 1231, 1230, 1229 (or however many days there are between now and April 8, 2014)>>>

So moving to Windows 7 is a big deal. Why?

Profiles
A profile is a profile is a profile, right? Unfortunately, we’ve been spoiled in all the years leading up to Vista. Profiles from Windows Vista on are called v2 profiles, and they’re incompatible with profiles from Windows XP and back. Vista has been out going on 5 years now, so why are we talking about this? That’s simple... nobody migrated to Vista.
So where does that leave us? What do we have to consider when migrating to Windows 7?

- What are the differences?
- How do we actually migrate the profile data?
- Misbehaving applications
- What if the users bounce between Windows 7 and Windows XP

The primary reason for profile incompatibility is that the folder structure has changed from v1 to v2. On the surface this is obvious because data is stored in C:\Users, and the so-called Special Folders are now located in the root of the profile (Pictures, Music, etc...), but it goes deeper than just the root of the profile. Try to find the "Application Data" folder, and you’ll see what I mean. It’s been renamed to AppData, but even that has been divided into more folders, like Local (the old <profile>\Local Settings folder that’s supposed to contain computer-specific settings), LocalLow (a new location for "low"-integrity apps to save settings, like IE plugins), and Roaming (the classic Application Data folder that contains user-specific settings that can travel with the user).
Another change that isn’t so obvious, though, is the way locales are handled with profiles.
In v1, profile data was stored in location-specific folders (like “Application Data” for English users and “Anwendungsdaten” for German) based on the local defined on the local machine. This posed a problem when roaming to machines with different locales because the settings would not transfer. v2 profiles fix this by removing the location-specific folders and storing all the data in the same folder no matter the locale. v2 profiles do utilize NTFS Junction Points, which are kind of like local symbolic links, for compatibility so that requests for location-specific folders are routed to the appropriate location.

Migration can be done with a tool available from Microsoft called the User State Migration Tool, which is part of the Windows 7 AIK. This will migrate your settings from v1 profiles to v2 profiles, which is great if you’re moving all the machines a user would access at the same time. The problem is, this doesn’t help poorly-written applications (like ones that hardcoded to a certain folder), or users that want to bounce from one OS to the other.
There's lots of tools out there that can help, though. AppSense, RES Software, Immidio, Liquidware Labs, Scense, and many other companies have come up with solutions for both doing the migration and for making the profiles portable. It's helpful from an OS migration perspective, but also with regards to those rogue apps. If you have a misbehaving app that only works on a machine with v1 profiles, you can put that app on a Windows 2003 terminal server or in an XP VM (client or server hosted) and the user could continue to use the same profile.

That means that, with a little help, your users can bounce between v1 and v2 profiles until all your systems support v2 profiles.
Security

Remember how User Account Control (UAC) was supposed to make everything more secure and add a line of defense between the user and the baddies? Remember how Vista was a little more aggressive than we wanted it to be with that whole permissions request thing? Thankfully, that part of the security system was fixed in Windows 7, and you don’t have to choose just “on” or “off” anymore.

AppLocker is an updated version Software Restriction Policies (which is still available), that lets you manage the applications a user has access to. It’s not an end-all application security solution, but as an admin you have the power to allow or restrict access to certain applications through the use of Group Policy.

x64

Remember, while x64 Windows can run most 32-bit apps no problem, there’s no 16-bit subsystem. So you’re S.O.L. for any 16-bit apps. “What 16-bit apps?” You might think. “Didn’t the world move to 32-bit apps almost 20 years ago?”

The answer to that is ‘yes.’
BUT...

While there aren’t that many 16-bit apps out there, there are still a lot of 32-bit apps with hidden 16-bit components. And often you won’t find these until you try and run them on an x64 edition of Windows and they just don’t work.

For example, what if your 32-bit app uses 16-bit ODBC components? Or maybe it’s a fully 32-bit app, except the super-old report generation engine is some 16-bit thing that the original software vendor licensed in the 1990s. Or maybe it’s a 32-bit app with a 16-bit DLL for the menu controls.

Or, in a specifically masochistic scenario, some apps are fully 32-bit no problem, but their installers are 16-bit!!! D’oh!! (So it will run fine.. it just won’t install. :)}
New Hardware
If you skipped Vista, lucky you! But now you’ve got an even deeper dependency on updated hardware than you had before. If you weren’t updating your hardware all along while waiting, good luck with that!

Actually, if you’ve got updated hardware, then business-as-usual is an easy step. You still might be a good candidate for desktop virtualization, but at least you’re in a more flexible position than...

...those that don’t. If you’ve got hardware to update ON TOP OF a new OS to deploy, you should be weighing the cost of a Win 7 migration and the cost of new hardware against a desktop virtualization project of some sort. Let’s face it - if the user has Windows 7 at home, complete with Aero goodness and fancy new applications, they’re not going to be happy coming to work with a different OS and apps on a shitty piece of hardware.

Don’t worry about all that desktop hardware, though - you can repurpose each machine as a thin client. (URL Shortener?)
App Compatibility

Once you start digging into Windows 7, you'll quickly realize that a bunch of your existing Windows apps don't work! So what can you do??

You've got several options

App virtualization

<<<<<Since we also talk about app virtualization elsewhere in the book, maybe you can draw an arrow that goes across a bunch of pages to the app virt section>>>>

While most people look to app virtualization to run conflicting apps side-by-side or to run apps by "streaming" rather than "installing" them, many app virtualization products can also be used to run older apps on newer operating systems. (For example, you can use ThinApp to run Internet Explorer 6 on Windows 7.)
Run your Windows XP apps in their own virtual machine

Another way to access Windows XP apps on a Windows 7 desktop is to run a full copy of Windows XP in a virtual machine on your Windows 7 host. Microsoft even provides a feature (based on Virtual PC) called “Windows XP mode,” a free download for Windows 7 that gives you a VM for running your older apps. (<<Shady alert: This is how Microsoft is able to claim Windows 7 has ‘100% compatibility’ with Windows XP>>.)

You don’t need to deal with Windows XP mode if you don’t want to though—there’s nothing stopping you from running Windows XP on VMware Workstation or one of the other desktop virtual machine solutions.

Terminal Server or VDI published apps

An easy way to deal with non-compatible apps might just be to run them on a different machine and access them from your Windows 7 host via a remote seamless window. A Terminal Server or VDI instance running your app with an icon in the Start menu of your Windows 7 client is a pretty easy way to get this done.
Update your apps

Finally, don’t forget that the simplest way to make an older app run on Windows 7 might just be to contact the vendor and get a new version of your app!

How do you know if your apps will be a problem in Windows 7?

The best way to figure out which apps will and will not work on Windows 7 is to just build a Win7 test machine and start installing your apps to find out! You can also get started with the Microsoft Application Compatibility Toolkit, a free download from Microsoft that contains components for automatically verifying and testing your currently installed apps.
Let’s talk about clients. Thin, fat, or mobile, Windows, OS X, Android, iOS, Linux, or BlackBerry...whatever you want, right? Any Any! YAY!

Man, there’s a lot of them, and I can’t imagine the amount of work these companies like VMware and Citrix put into making client packages for all of these devices and form factors. It’s nice to know that you can access your virtual desktops from just about anything, but how do you choose your endpoints?

Choosing your endpoint strategy probably deserves as much consideration as you put into any other aspect of your environment. I’d bet it’s one of the first virtues of desktop virtualization that you learned of. Who hasn’t heard “You won’t have to manage PC’s anymore because the desktops will be in the data center and they can just use thin clients to access their desktop,” before? If only it were so easy!

Here’s what I consider when trying to decide an endpoint strategy:
- OS (Windows, OS X, Linux, mobile, whatever...)
- Managed/Unmanaged devices
- Local Capabilities
- Vendor Reputation
- Completeness of system
- Technology
- Price
Your first step is to prioritize that list, because if you absolutely require the best local capabilities, you’ll want that to come above all else. That’s why price is at the bottom, because you need to learn your requirements before you can find the right device. THEN you can worry about price.

I put OS first, because there’s plenty of people that feel like Windows is the best thin client OS, even if it’s there for no other reason than to run the VDI platform’s client. Most people that choose Windows on the endpoint do so because it has the largest support base. Windows has been designed to run on any typical PC or laptop, and all the vendors have Windows clients that are full-featured with every bell & whistle available. Essentially, the user experience will almost certainly be the best when using a Windows client.

The problem with Windows is that the ENTIRE POINT of doing VDI is to get the desktop off the user’s desk! Leaving Windows there means another box that has to be managed, patched, and secured. To do that best, you probably have to leave it in the domain, which means you’ve doubled your number of computer accounts in AD. I don’t know if that’s so bad, but I loves me a clean AD.
Does that mean Windows is a bad idea? Not necessarily, but if drastically reducing endpoint management is high on your list, you'll likely find that Windows is probably not the best idea. There are a few products out there like ThinLaunch's ThinDesktop and triCerat's triShell Kiosk Edition that allow you to run Windows in a locked down way so it can be treated like a thin client, but that doesn't necessarily help with management, patching, or overall security.

So now you've sworn off Windows, right? If not, that's ok. You're not alone, but if you did it to have apps installed locally instead of in your VDI desktop, you should look at Application Virtualization or RDSH solutions. If you can get all application execution to be remoted or isolated, that makes your desktop management a lot easier.

Now let's say you've decided to skip PCs running full on Windows, but now you're sitting on this pile of hardware. Do you ship it off to your DR site? Probably not, since you'd still have Windows boxes to manage. Do you trash it? No, you don't. That's illegal. You could recycle it, but even that will cost you money. You could turn that hardware into thin clients, though.
There's quite the market for software that will convert your existing hardware to thin clients. There's even a Windows solution called Windows ThinPC that comes with many of the headaches of maintaining Windows on the endpoint, but without the flexibility to make it worthwhile! Wyse, HP, DevonIT, and Igel all have products that replace Windows and allow you to manage your converted hardware alongside their thin client hardware. These are great because they allow you to transition your hardware at your own pace instead of overnight.

For you uber nerds, there's also a fair number of Linux-based solutions in the OSS world aimed at the same thing, although support is sometimes a little shaky.

Now I said it. "Thin client hardware." Damn, there's a lot of them, and their capabilities are so different. You need to figure out your requirements before you start, and part of that is determining WHY you want to use thin clients over, say, a repurposed PC-based solution. Then, you've got to figure out what you need to get out of your thin clients. Do you need to support peripherals? multimedia? WAN access? multiple displays?
Asking and answering those questions (talking to yourself is ok) will help you choose the right solution by helping you eliminate the systems that don't meet your needs. For instance, if you require PCoIP to perform at its best, you can narrow down your choice down to only the devices that have the hardware decoder chip in them.

Everything we've talked about so far requires some sort of management. Even if it's not as intense as Patch Tuesday-type stuff, all of the devices have configs and management consoles and such. But there's another type of client that doesn't require much, if any management - zero clients.

A zero client is one that stores nothing. No OS, no data, no configuration. When they turn on, they get in touch with a centralized system and receive their OS and their configuration. Wyse, HP, Panologic, and, of course, Oracle (via Sun) have zero client solutions that are shipping today, but each is different and you should check them out to see if they fit your needs.
What about my fancy-pants mobile device?

Story time: One day, some bigwig boss person comes in and says, "Hey, I'm headed on the road, and I don't want to take my laptop even though it's a super-light MacBook Air and you already spent a week loading Windows on and getting all my apps and data on it. Can you make my new LufLogix Wonder Tablet work with our VDI?"

"Sure," you say, but deep down you know this won't end well. Within minutes, you have the device set up, and the bigwig happily goes on his or her way.

As fun as it is to be able to access Windows desktops from anything, anywhere, you probably can guess how this story ends. These devices weren't made for Windows, and Windows wasn't made for these devices. The interface is clunky, at best, and no matter how good the client is, it will NEVER be as good as a thin client or Windows client.

More than one desktop virtualization project has died due to some important decision-maker having a bad experience with it, which is all the more reason to put a lot of consideration into choosing the endpoints that you support.
Let’s face it, though— you can’t avoid the mobile devices. I have clients from every manufacturer on my iPad, and I use maybe two of them, and only then for remote support. I’ve never used them for real work, but they’re there just in case.

THAT is the main point about these mobile devices. They make great secondary clients that help in a pinch or in certain situations, but they don’t make very good daily drivers. Nobody bought an iPad and threw out their PC. Well, nobody should have :) 

Had enough endpoint talk? Here’s one last point (my “end” point—ha!) –
THE ENTIRE GOAL SHOULD BE TO MANAGE ONLY ONE DESKTOP PER USER.

By “desktop,” I mean the abstract desktop—the one in the data center, not the physical device under the desk. The ideal situation has each user’s desktop follow them from one device to another. Multiple devices are unavoidable, so do yourself a favor and reduce the management of those devices as much as possible. If you can provide One Desktop Per User AND reduce management of the endpoints, you’ll have yourself the makings of a well-oiled VDI.
You need a bare metal client hypervisor like you need a hole in your head!

Flat out, when VMware canceled their CVP program, I thought it was a huge mistake. I thought it was crazy. I thought it meant that they were going down hill. I thought Citrix was doing the right thing. But you know what? The idea of the client hypervisor. the sexiness. man it just doesn't jive with reality.

A few years ago I wrote about a paradox: "If we had the technology to make client hypervisors work, then we wouldn't need client hypervisors."

And that's still true today!

A Type 1 hypervisor is where the hypervisor is the base OS. It's installed onto bare metal, and everything else runs as a VM. Examples of bare metal client hypervisors are Citrix XenClient and Virtual Computer nxTop.

A Type 2 hypervisor is where the hypervisor is installed like any other application on top of an existing OS. Examples of these are VMware Workstation, VMware Fusion, VMware View Client with Local mode, Microsoft Virtual PC, Parallels Workstation, Microsoft Windows XP mode, etc.>>>
Why are bare metal client hypervisors so challenging??????

I swear. When I first saw ESX in 2004, I just couldn’t wait for that to come to the desktop. I had been using VMware Workstation for awhile, but I felt that it was a bit clunky and kind of confusing sometimes. But if they ever came out with a bare-metal hypervisor for laptops... man! Imagine how bad ass that would be!!?! We could use one image for every user! The user interface would be simple! We could have easy ways to patch, backup, and encrypt our systems! It would be an amazing nirvana!! <<<<image about me fantasizing here>>>

But when the first bare metal client hypervisors started coming out five years later, my fantasy bubble popped. :( It turns out that client hypervisors weren’t nearly as cool as I thought they’d be.

First of all, it became clear that the reasons that server hypervisors became successful were not really applicable to client hypervisors. Servers live back in the datacenter, and the only user interaction is via the network. Clients are physically, locally, and interactively used by users every moment of every day.
First of all, it became clear that the reasons that server hypervisors became successful were not really applicable to client hypervisors. Servers live back in the datacenter, and the only user interaction is via the network. Clients are physically, locally, and interactively used by users every moment of every day.

And virtualizing servers is easy because you only have to virtualize four resources: CPU, memory, disk, and network. Clients have all that, plus graphics cards, GPUs, sound cards, webcams, multitouch trackpads, USB devices, Firewire, lid-closure switches, WiFi, battery power status, webcams, thunderbolt, fingerprint scanners, SIM cards... the list goes on.

So to make a “really good” server hypervisor, you only have to address those four areas. But if you only address those four areas with a client hypervisor, you’ve got a really bad experience!

So how do the vendors get around this?

Should they not release a Type 1 client hypervisor until they get all that worked out? If so, then we probably won’t see a client hypervisor until 2035.
Should they just "open up" their client hypervisor so that those "Core 4" items are virtualized, but that the rest are not? That's no good, because you won't be able to use the same VM on machines with different hardware, which kind of negates the whole point.

One of the advantages of VMs versus locally installed OSes is that VM images are easily "portable" between physical hosts. So when it comes to client hypervisors, people think "Oh cool, so now I can have just one image for all of my different laptops."

WRONG!

Let's take the GPU for example. If the hypervisor vendor doesn't virtualize the GPU (and none do today), the best you can do is "pass through" which means that the GPU will work in the guest VM, except that you also have to install drivers in the guest for that specific GPU.
Brian: But I have all different laptops. Some have Intel GPUs, some have AMD/ATI, and some have Nvidia?

Vendor: So?

Brian: So I can't have one image that I can share across all three types of hardware.

Vendor: Sure you can! Just install all three driver sets in your VM. Then it will work everywhere.

Brian: But couldn't I have done that anyway?

Vendor: ...

Brian: Why do I need a client hypervisor to do that?

Vendor: ...

Brian: Tell me again why I'm using a client hypervisor?
The bare metal client hypervisor vendors talk about a lot of advantages of bare metal hypervisors running on clients. They'll say things like with a client hypervisor, Windows updates are as simple as disk image updates. And backups are as simple as incremental disk image backups. And that when using client hypervisors, you can easily encrypt your disk images and enable remote “kill pill” capabilities.

And the list goes on, which leads back to the paradox. <<<maybe a big arrow pointing up to the paradox on the previous page?>>>

To be honest it’s taken a few years to realize that client hypervisors are just a tool for managing laptops. Client hypervisors don’t give you magic powers to fix Windows, they just help you deliver updates. Client hypervisors don’t give you a magic way for your image to be cross-system compatible. they just provide a way for your cross-system compatible images to be updated. Client hypervisors don’t give you a magic solution to encrypt your disk images. they just give you another way to encrypt your disk images.
So knowing that, why would anyone use a client hypervisor?

It’s true... instead of using a client hypervisor, you can build a good image, deploy it with some client management or image management software, encrypt it with some disk encryption software, add remote “kill pill” capabilities with some security software, enable continuous backups with some backup software... All of these capabilities are available on their own.

Or you can just use some bare metal client hypervisor software.

Which is better?

Ford or Chevy? Baseball or Cricket? Boxers or Briefs? <<<Maybe some doodles of these>>>
Really it doesn't matter. To each, his own!

But that's the key takeaway here. Client hypervisors are not a magic bullet. They're just a nice way to manage your endpoints. Should you use them or traditional management tools? That's up to you.

<<<SIDEBAR somewhere in here>>> Maybe a quick note like "Client hypervisors get a free pass to cooltown because they use virtualization to solve an age-old problem, but they do so with a hot technology. But that doesn't mean there aren't other ways to achieve the same goals.>>>
A lot of people think about app virtualization only in the context of desktop virtualization, but actually app virtualization can be used a lot in traditional physical desktop environments. In fact even if you only use desktop virtualization for a small percentage of your users, you can use app virtualization for everyone.

<<<DOODLE: APP VIRT FOR EVERYONE!!!>>>

In fact this is a good way to start to apply some of the advantages and concepts of desktop virtualization, but in a slow and controlled way. You've got to go for the small, easy wins.

The problem is that everyone tends to think of app virtualization as a way to offer a kind of "dynamic personalization" on top of a shared, locked-down disk image (like for VDI or Remote Desktop). They think "Oh, so all my various users will share a locked-down image, and then I'll customize it for each user by delivering virtual apps into their VM after they boot up."

And then people think, "Hmm.. but if I'm not using desktop virtualization in a big way right now, I guess I don't need app virtualization?"

WRONG!!!!!
Think about your existing "pre-desktop virtualization" environment. You have lots of traditional desktops and laptops out there, with most probably running Windows XP. And you probably also have dozens (if not hundreds) of different apps to support. Now if you're like most IT Pros today, you're more focused on getting to Windows 7 than getting to desktop virtualization... and that's fine. But don't think that you should just ignore app virtualization.

In fact, app virtualization can help you get from Windows XP to Windows 7, even if you don't do any desktop virtualization at all! After all, any existing physical app that you virtualize today is one less app that you have to worry about in your new Windows 7 environment. (And if you eventually do go to desktop virtualization, it's one less app for you to worry about there too.)
There are a ton of app virtualization products on the market today, including:

- VMware ThinApp (formerly Thinstall)
- Microsoft App-V (formerly Softricity)
- Citrix XenApp Streaming
- Symantec Workspace Virtualization (formerly Altiris SVS)
- InstallFree
- Spoon (formerly Xenocode)
- Endeavors Technologies

So which one of these products is best? Unfortunately none is able to virtualize 100% of your apps, and even if you combined all seven you probably still couldn't virtualize 100% of your apps, but that doesn't mean that app virtualization is worthless!!!

Even if you only use app virtualization to virtualize 10% of your apps—hey, that’s 10% fewer apps that you have to deal with the old way! :)

Brian's top Use Cases for App Virt
- Migration from XP to 7
- Migration from physical to virtual desktops
- Use for apps not in base image
- Isolate to avoid native install conflicts
- Hide your tracks for adult websites that require IE6
And remember that you can integrate these app virtualization solutions into your existing methodologies for desktop management, whether that's automated deployment systems like Microsoft System Center or Symantec Altiris Client Management Suite, or simply manual application installations by desktop support folks.

If you don't have budget for desktop virtualization today, you can make a nice dent in your desktop virtualization project of the future by starting to virtualize your apps today. And if you're good, you can get today's app virtualization products included as part of your Windows 7 deployment budget.
Ahh.. the network.

In the old days before desktop virtualization, we could happily ignore the network. (Well, maybe we couldn't fully ignore it, but we could rest assured that the general performance of our desktop and the general speed of the network were in no way related. It was a simpler time... :)

But now... man!! So many of these various desktop virtualization technologies really rely on the network. And this sucks for us because we used to be "desktop people" or maybe "application people..." .. but the network.. man, that was someone else's domain.

But now we've learned quickly that a "bad" network (whatever that means) equates to a "bad" user experience, which also means a bad day for us.

Bummer.

So let's dig into this.

I guess the first thing is to understand that different styles of desktop virtualization use the network in different ways.

Fundamentally, you can categorize the various types desktop virtualization technologies into two broad groups:
Network usage for remoting protocols.

When we're talking about "datacenter-based" desktops and "the network," we're really talking about the performance of the various remoting protocols. Everyone wants to know: Which is best? Is PC-over-IP better than HDX? Can I just use RDP? What about those third-party accelerators?

I put PC-over-IP and HDX to the test for a month each earlier this year. I used them all day for anything work-related, and after my completely subjective, only one user for an entire WAN link, entirely qualitative testing, I can emphatically say that they're both fine. I wouldn't pick a product based solely on the remote protocol in my specific situation. But my situation is not your situation, and THAT is what this section is all about.

Let me say this: There's a lot of bad information out there! (maybe a diagram of something smelly or bad?)
Let's first talk about bandwidth.

Q: How much bandwidth does a particular remoting protocol need?

A: As much as you can give it! (Seriously!)

This is one of the first things that everyone wants to know. How much bandwidth does xx protocol take?

The answer, of course, is that it depends. What’s the user trying to do? Edit videos or just type documents in Word? Is the user surfing the web? Is their audio? What about client-side webcams and USB drives? How many screens does the user have? What resolution? Is the user’s session 32-bit or 24-bit color?

Based on all of these factors, I can safely say that you can safely plan for your users to consume anywhere between 6kbps and 100mbps.

Any questions? :)

Seriously, there’s just too much that goes into the bandwidth calculations, and it’s just too easy for vendors to fudge the numbers. (Like they might say that a certain bandwidth provides an “acceptable” experience for a user, but you might personally find that their version of “acceptable” and your very own “acceptable” are not the same thing.)
So the next thing that usually happens is that people start testing. They’ll build a lab environment with a remote VDI desktop and a client and then they’ll make a remote connection. Then they’ll perform some usability tests, perhaps with PC-over-IP, HDX, and RDP, all one after the other. Then they use perfmon to see which one consumed the least bandwidth.

And guess what? ... THAT TEST IS TOTALLY WORTHLESS!!!! DO NOT WASTE YOUR TIME!!!!!!!!!!!!!!!!!!!!!
Looking how a protocol (any protocol, not just remote display) uses bandwidth in an unconstrained test environment is in no way indicative of how that protocol will perform in a bandwidth-constrained environment. For remoting protocols, you cannot extrapolate a single user session out to guess how multiple users will behave.

"User experience" and "bandwidth consumed" are totally different dimensions.

Digging into these points a bit more, if you have an unconstrained environment, looking at bandwidth consumption is irrelevant. A good protocol will use as much as it possibly can. More data across the wire means a better user experience and less load on the server and client with all the fancy compression and caching tricks. So if I’m a remote protocol and I see a wide open highway, I want to let ’er rip! In fact I would say it’s a bad thing if a remote protocol doesn’t use more bandwidth when the network becomes unconstrained. It’s like I want to say, "Ummm, hello? Dumbass? There’s a wide open network here, so can you please take advantage of it?"

So when the network is unconstrained, that remote protocol better deliver an experience that’s f’ing perfect, because if it doesn’t, that means there’s some problem with the protocol’s ability to scale up.
That said, the unconstrained environment is not realistic for most real-world use cases. A better test would be with multiple users over a single connection. (Maybe with tests for LAN and WAN.) And that goes to my second point. Too often people try to figure out an "average" bandwidth consumption for a remote protocol. Sure, it is possible to work out what the average consumption was per user per second per connection, but it's really, really difficult and inappropriate to try to apply these averages to other environments.

Does that mean that each user averaged 70k? Yes.

Does that mean that a single user will have the same quality of experience over a 70k connection? No.

Does that mean that 20 users will have the same quality of experience over a 1400k connection? No.

Does that mean 5 users will have the same quality of experience over a 350k connection? No.

I love Shawn Bass'es quote about this: "The average temperature in the desert is 70 degrees, sounds great, right? Sure, except that 'average' comes from it being 135 degrees in the day and 15 degrees at night."

Yup.
What about Client-based desktops?
Say you think this is all too complex, and instead of putting the VMs in the data center, you want to just put the VMs on the client side. There’s still benefits there of portability, backup and such, but all this network consideration talk goes right out the window, right?

Nope.

The problems are generally not that new, but they’re still there. When a user with a roaming profile logs in, that profile data is pulled across the network. Even if they just use mapped drives and folder redirection, that data is still coming across the network.

But there are new problems, too! Take backups, for instance. If you’re backing up the entire VM, that’s a significant amount of data to consider, and it, too is subjected to the network conditions.

On an unrestricted network, it’s no big deal, but with branch offices and remote users, it’s enough data that you need to take a step back and learn exactly what is going on between your clients and the datacenter. Yes, running the VM locally might solve the problem of having a bad experience over a crappy WAN link, but it might break some other features that you were counting on using.
Desktop virtualization is not server virtualization! This means that storage for desktop virtualization is not the same as storage for server virtualization.

Think about it. When you pick a server hardware standard for your company, that’s what you run in your datacenter. But when it comes time for desktops, do you plop a 1U server on the floor under each user’s desk? Of course not! You give each user an actual desktop or laptop.

So the same thing is true for virtual storage. The needs of servers and desktops are different. So I don’t care if you have an awesome EMC or NetApp whatever for your virtual server infrastructure—when it comes time to look at storage for virtual desktops, you’d better damn well make a separate decision. If you just “copy and paste” what your server people did, you will fail.
When it comes down to thinking about storage for virtual desktops, there are really two issues to deal with:

- Performance
- Image management

In some ways these two issues are independent and unrelated, although in some cases the decisions you make for one can impact the other. Still, you have to treat them as two separate issues really.

**PERFORMANCE**

When it comes to performance, faster is (1) better and (2) more expensive. So really you’re balancing that. If you wanted to guarantee a well-performing storage subsystem for your virtual desktops, I would say if you spend $5k per user on storage then you’d get a pretty fast solution! (Unfortunately you’d probably also get fired since that’s way too expensive for most scenarios.)
But if you want to get your storage costs down to hundreds-of-dollars per user, then you'd going to have to look at a bunch of different options.

*Physical storage appliances
Lots of new vendors are out there offering appliance-based storage, like XtremIO, Nimbus,

*Storage integrated into server platforms
Some vendors are now offering storage & server processing built-in to the same platform, like V3 Systems, Pivot3, Nutanix, and Fusion-io

*Virtual storage and software-based storage solutions
Software-based virtual storage solutions, like Virsto, Atlantis Computing, Datacore
IMAGE MANAGEMENT

People used to think that image management and storage were the same thing. And I guess in some ways they are. I mean if you decided that your “image management” solution is going to be that all your users just share a single common master image, then I guess you could also say that solves your performance and storage issues (since that single image can be cached and shared across loads of users).

But maybe you want your VDI users to each own and maintain their own personal images. Is it at all possible to let each user have their own image and to be able to provide that image for less that $5k a user?

The answer is yes, I think. If not today, then soon.

There are a lot of third-party (and even a few built-in) happy mediums in the desktop virtualization space that give users the experience of each having their own image while allowing multiple users to share the same base image. That’s great for performance, and gives you added flexibility if you need it.
A note about “storms” (with doodle of storm?)

A lot of people talk about how your virtual desktop storage has to be robust enough so that it can support “storms.” Well it doesn’t out that there are actually two (maybe 3) completely different kinds of storms... one that’s easy to defend against, and one that’s difficult.

The “easy” storm is that “boot storm,” so called because it happens when a lot of computers are booted at the exact same time. But that’s an easy problem to solve--especially with desktop virtualization! (Just configure your desktop VMs to auto-boot in waves in the morning.) Plus, booting a computer is a read-intensive activity, and the data read is largely the same from system-to-system, so it’s an easy thing to cache.
The real storm problem is that “logon storm,” which happens when a lot of users log on at the same time. This is tough because logons are write-intensive, which is inherently more challenging to cache, and it’s different from user-to-user. Also, logon storms happen when actual human beings logon. So while it’s easy to pre-boot your desktops in the wee hours of the morning, it’s not like you can really tell your employees not all come in between 8 and 8:15! So you need to make sure that the storage system of your virtual desktop environment can handle this!

The third “storm” is one that can be avoided with common sense—the antivirus storm. AV software on your physical desktop under the desk is no big deal, but if you load AV into all of your VM’s, you’re going to be in trouble when every machine starts doing a scan. An AV storm is easy to avoid, but it involves changing some old habits. Some orgs just turn it off altogether, but if you were worried about viruses before, you should worry about them now, too. There are out-of-band AV solutions, like VMware vShield Endpoint or McAfee MOVE, that operate in a separate security VM and can intelligently keep watch on virtual machines from outside the
Desktop virtualization is not server virtualization! This means that storage for desktop virtualization is not the same as storage for server virtualization.

Think about it. When you pick a server hardware standard for your company, that's what you run in your datacenter. But when it comes time for desktops, do you plop a 1U server on the floor under each user's desk? Of course not! You give each user an actual desktop or laptop.

So the same thing is true for virtual storage. The needs of servers and desktops are different. So I don't care if you have an awesome EMC or NetApp whatever for your virtual server infrastructure--when it comes time to look at storage for virtual desktops, you'd better damn well make a separate decision. If you just "copy and paste" what your server people did, you will fail.

When it comes down to thinking about storage for virtual desktops, there are really two issues to deal with:

- Performance
- Image management
Q: What's the best way to minimize the amount of disk IO for your desktops in the datacenter?

A: Simple! Don’t put your desktop in your datacenter. Remember that in addition to VDI and Remote Desktop, you could run your desktops locally on your client hardware, either with a client hypervisor, a streamed OS, or a simple old-fashioned local desktop!
But maybe you want your VDI users to each own and maintain their own personal images. Is it at all possible to let each user have their own image and to be able to provide that image for less that $5k a user?

The answer is yes, I think. If not today, then soon.

There are a lot of third-party (and even a few built-in) happy mediums in the desktop virtualization space that give users the experience of each having their own image while allowing multiple users to share the same base image. That’s great for performance, and gives you added flexibility if you need it.

<<<SIDEBAR A note about shared images with VDI>>>

A lot of vendors are pushing the ability to “share” a single master locked-down image across multiple users. VMware Linked Clones, Citrix shared images, NetApp Flex Clones, etc... But I have to say, if your VDI environment is simple enough that your users can share the same base image, then isn’t Remote Desktop Session Host also an option (since that kind of is a shared image desktop solution)? Whether it’s the best option or not depends on the use case, but it’s still an option.
A note about “storms” (with doodle of storm?)

A lot of people talk about how your virtual desktop storage has to be robust enough so that it can support “storms.” Well it doesn’t out that there are actually two (maybe three) completely different kinds of storms... one that’s easy to defend against, and one that’s difficult.

The “easy” storm is that “boot storm,” so called because it happens when a lot of computers are booted at the exact same time. But that’s an easy problem to solve—especially with desktop virtualization! (Just configure your desktop VMs to auto-boot in waves in the morning.) Plus, booting a computer is a read-intensive activity, and the data read is largely the same from system-to-system, so it’s an easy thing to cache.

The real storm problem is that “logon storm,” which happens when a lot of users log on at the same time. This is tough because logons are write-intensive, which is inherently more challenging to cache, and it’s different from user-to-user.
Also, logon storms happen when actual human beings logon. So while it's easy to pre-boot your desktops in the wee hours of the morning, it's not like you can really tell your employees not all come in between 8 and 8:15! So you need to make sure that the storage system of your virtual desktop environment can handle this!

The third "storm" is one that can be avoided with common sense--the antivirus storm. AV software on your physical desktop under the desk is no big deal, but if you load AV into all of your VM's, you're going to be in trouble when every machine starts doing a scan. An AV storm is easy to avoid, but it involves changing some old habits. Some orgs just turn it off altogether, but if you were worried about viruses before, you should worry about them now, too. There are out-of-band AV solutions, like VMware vShield Endpoint or McAfee MOVE, that operate in a separate security VM and can intelligently keep watch on virtual machines from outside the machine so you can avoid AV storms and keep the same level of protection.
What about Active Directory?

What's nice about VDI is that there's nothing to change with AD, really. At least nothing that you're not already worried about (domain vs. local users, user containers, computer containers, global/local/universal groups, and so on). The biggest change is that, essentially, the SID has been deemed irrelevant, and you no longer need to change it. That's why VDI platforms save time using a technique called "quickprep" instead of Microsoft SysPrep.

Wait, what?

You don't need to change the machine SID anymore? That's ridiculous! Crazy! Cradiculous! We've been doing that for 15 years now and all of a sudden it's no big deal?

I guess we never had to do it, actually—at least according to Mark Russinovich, who pretty much owned the SID-changing space until he called it all a myth. I guess if he can change his mind, so can I. Still...WTF? Call it Job Security

Quickprep vs SysPrep

"Quickprep" and SysPrep are both tools for genericizing an OS image. Sysprep a tool from Microsoft that uses a two-part process that involves first clearing any unique information from the machine, then, at next boot, regenerating all unique information through a mini-setup process.
Quickprep was created because SysPrep is freaking slow, and when a user wants to connect to their non-persistent VDI image, they want it to start up immediately, not wait a few minutes while Sysprep does its thing EVERY TIME.

It takes so long because Sysprep removes unique identifiers from the machine, including the machine SID. It also changes the SUSClientID used by WSUS for updates, and the CMID (Computer Machine ID), which is used for tracking licenses via KMS. It then re-packages the entire OS for a fresh installation. Answer files can automate the Sysprep mini-setup process, but it still takes longer than people want to wait.

The problem with Quickprep is that it does essentially the bare minimum to make an image generic, which some think could lead to problems down the road. So far, Quickprep solutions appear to be working just fine, except for the fact that they're not supported by Microsoft. At this point, only SysPrep is supported for cloned machines.
I heard this example once from a guy name Rick Mack. RMIT™ - Rick Mack is the man He set up three servers, all clones of each other. The first one was promoted to domain controller, and the other two were File/Print and Management servers. Local & Domain admins worked ok, but anyone else logged in to the management or File/Print servers had weird stuff happen. "Unable to contact IP driver" What the hell is an IP driver? *HELPFUL* *FAIL*

Anyway, SysPrep fixed it! So much for SIDs not mattering, right? But why?

Turns out when the first machine became the DC, the machine SID became the domain SID. So now the Domain SID is the same as the machine SID for the other two machines. YIKES!

Now consider the RID - because that's the last part of the domain SID.
The RID is unique and is assigned as the object is created. It, combined with the Domain or Local ID means that there is no (or at least a ridiculously small) chance that two objects will have the same exact SID, except when cloning is involved.

The Administrator RID is always 500 (anything under 1000 is there by default), which is why logging in as admin worked. The SID for local admin was identical to domain admin and identical to the machine SIDs on the other servers! But when the other users tried to log in to the domain, the local machines saw a SID with the same unique identifier as themselves, but it can't find the user in its local SAM Security Account Manager because it doesn't exist there (and it shouldn't!). Still, it tries to assume the user is legit because it has the proper SID, and the weird stuff starts happening.

Sysprep fixes it because it changes the local machine SID, which means when a domain user tries to log in, the local machine can tell the difference between a domain user and a local user and forward the authentication request on to the domain controller.

WHEW!
So, back to VDI. It sure sounds like SIDs matter, still, right? \textit{NO!!!} It's ok if all of your virtual desktops have the same SID... here's why:

That example above is a fringe case that involves domain SIDs, and when I say SIDs don't matter, I mean \textit{LOCAL MACHINE} SIDS, not domain SIDs. You'd never clone a machine after it joins the domain because you know that causes issues... this is different.

Local machine SIDs can be the same \textit{AND} still be secure. If one machine is accessing another, the remote machine (the one being accessed) handles authentication (\textit{SAM} for local accounts, \textit{AD} for domain accounts), and the local machine is just sending along its information like normal. The remote machine doesn't care one bit about the SID on the local machine... all it cares about is that the account exists in the remote machine's \textit{SAM} or in \textit{AD}.

The only time it can bite you in the a$$ is if local accounts are duplicated on other machines and the password isn't kept secret, but \textit{THAT} is a problem whether you clone or not.
Almost convinced? What’s left? What about SCCM? WSUS?

It’s true that cloning a machine can screw those up, but it’s not because of the SID. Each of them use their own GUID, NOT the SID, to identify machines on the network. It’s almost like MSFT planned it this way because they knew the local machine SID was unreliable or irrelevant as a true unique identifier. (ya think?!) 

For SCCM, you can’t install the client before you clone, or the identifier (called an “SMS GUID”) will be the same for all machines.

"WAIT!" you might say, “The client will change its ID if the hardware changes!"

In a traditional world, that’s true, but in a virtual world there’s no guarantee that the hardware will change enough for SCCM to change the ID. When doing VDI, you need to wait until AFTER you create the image before deploying the SCCM client.
For WSUS, it's not that simple. The WSUS client is built in to Windows, and it has it's own GUID called the SUSClientID that, if cloned, stays the same on all your cloned machines. The only ways to work around that on cloned machines are to find a way to update the SUSClientID when a new machine is created or Sysprep the images as they're deployed.

So all this time it didn't matter? Crazy! Still don't agree? Bing Google for “The machine SID Duplication myth” by Microsoft's Mark Russinovich.

The bottom line? Stop worrying about the machine SID. If you have to worry about a SID, focus on the Domain SID. VDI + SysPrep is almost foolproof, but takes forever to build the image because of all the extra work it's doing. VDI + "quickprep" is ok in almost every situation, but you have to take special care to change the SUSClientID when using WSUS. In both cases, if you're using SCCM, you need to install the client after the image is deployed.
One of the main selling points of VDI is that it increases security, right? Like since VDI desktops run as virtual machines in the datacenter, no actual data crosses the network or leaves the data center. And if a laptop is stolen, you don’t have to worry about disk encryption because there’s no chance that any actual data would be left behind. Back in the nineties, Citrix used to call this “eyes-only” security.

I wonder if anyone remembers that I actually gave a presentation on this topic way back at iForum 2003. I talked about how server-based computing (which includes VDI) can increase security. In addition to eyes-only security, I mentioned benefits like the ease of locking down systems, the ease of patching, the absence of real data on the wire to intercept, and that everything can be encrypted. All this sounds good, right?

NOT SO FAST!!!
If you’re not careful, implementing VDI can actually make your overall environment less secure!

But how? The short version is that VDI moves your unpredictable users from out in the field into your data center. This can harm security in several ways:

Many organizations allow their desktop users to run with administrative rights, which is bad because users can install whatever they want (secure or not) into their Windows sessions, and many viruses are able to do more damage when users are logged on as admins. (Heck, a local admin with the ability to use Google can completely remove all group policies from his or her Windows instance... and that's running in your datacenter! Yikes!)

So in general, IT organizations approach security by as much as they can and then looking for anomalies. But to a security system, many desktop users doing many different things are by definition an anomaly. From the security system's standpoint, it's difficult -- if not impossible -- to tell what's the anomaly and what's OK.
Some people think, "oh, that's ok... we're just going to use "shared" disk images for our VDI, so we can just "revert" the images back to the baseline each time the user logs off. That means that we don't need to worry about security and antivirus software isn't necessary.

Yeah.... good luck with that!

Since the VDI instance is running in your data center, the user's desktop "playground" is on a trusted network from a trusted connection, even though it's not known what the user could do within his or her own session.

So how do you address these potential vulnerabilities and prevent your VDI environment from affecting the your network security?
First, you should never let your users run with admin rights, regardless of whether they are VDI or traditional desktop users.

Second, it's important to treat virtual desktops just like traditional desktops in terms of security and management tools. If you run antivirus apps on traditional desktops, run it in your VDI environment. If you use policy enforcement and lockdown tools in your traditional environment, use them in your VDI environment. The same also applies to patching, auditing, change control, etc.

Of course this can be difficult in a VDI environment. For example, with traditional desktops, each desktop has its own processor, and whether one desktop is maxed out or not has no impact on the other desktops. But obviously with VDI, this is not the case.

So, for example, imagine that you run full AV scans every day at the same time. That's fine for physical desktops. But for your VDI environment, you'll bring that whole thing down if you they all start running AV scans at once (that's called an "AV storm").

<<<<maybe a doodle with A's and V's falling like rain?>>>>>
So how can you deal with that? Well, a very manual solution is to just configure every AV scan to run at different times, or even turn off AV altogether. Ignoring AV is probably the worst thing to do, though, and there are other methods that involve managing AV outside of the desktop VMs. With solutions like VMware vShield Endpoint, a security VM running on the host has insight into each of the desktop VMs and can scan it intelligently to avoid creating undue load on the VMs.

Finally, remember that when you use VDI, you bring your users closer than ever to your trusted inner network. If you use virtual LANs or firewalls to segment servers from desktops, make sure that the VDI environment is on the desktop side of that firewall (even if the cabinet is in the same data center as your servers).

Remember, VDI users are desktop users, not server users!!!
Imagine a world without Windows...

I know it’s been said a lot, and there’s a Mark Twain-style joke in there about rumors of its death being greatly exaggerated. But I think now I can finally see the path to a world without Windows.

Why does this even matter? Because for the past, what, ten years, when we say "desktop virtualization," we really mean "Microsoft Windows desktop virtualization." In the context of IT, a "desktop" means a "Microsoft Windows desktop."

But that’s starting to change. The popularity of non-Windows device OSes, like iOS, Android, and macOS X,

For example, look at the people who try to use a Windows desktop on an iPad. (And of course there are a lot of ways to do this: VDI, LogMeIn, GoToMyPC, VNC, etc.) No one really wants to run *Windows* on an iPad. (Windows needs a keyboard and pointing device—not a five finger multitouch screen.) The only reason they do this is because they need to use a Windows app or because they need to access some data from their Windows desktop. And since remotely connecting to Windows is the only way to do that, that’s what they do!
But in the case of an iPad connecting to a Windows desktop, the iPad’s “desktop” is the iOS software installed locally on it. That’s where the user’s settings, connections, apps, and data are accessed. The Windows desktop is “just another app.” (And as more and more device-native apps are released, users will use their remote Windows apps less and less.)

Windows app makers realize that their keyboard-and-mouse apps don’t make sense on touch-based devices, which is why every day we see more and more apps written for native non-Windows devices. And as time goes on, we’ll see that the only Windows business apps out there are in-house, legacy, and few-and-far between.

<<< SIDE BAR Microsoft can add whatever multitouch shell they want onto Windows 8, but that doesn’t change the fact that today’s Windows apps are designed to run with a keyboard and pointing device. And the idea that companies will “redevelop” their apps for Windows touch is ludicrous. If they were going to redevelop their apps, they would just do it for iOS or Android.>>>
So what’s this have to do with desktop virtualization?

I’ve said this hundreds of times over the years: If you think about it, what exactly is a desktop? It’s a collection of applications. It’s a collection of settings. It’s a collection of data. It’s a local user interface and OS for the client device. It’s a container that policies and security can be applied to. It’s an application operating environment.

And for years, all of those roles have been fulfilled by a single product: Microsoft Windows. And all of the “desktop virtualization” thus far has been about running Microsoft Windows in different places and in different ways.

But what if instead of just looking at desktop virtualization as a way to run Microsoft Windows in different ways and in different places, what if we used desktop virtualization to break up all the different roles of Windows into separate components?

- Data could be stored in the cloud (public, private, or personal) and accessed from whatever device the user was on at a given time.
- Applications could run on any platform (locally or remote) and be delivered to the device.
- User settings could live in the cloud and transcend local OS and device settings.
- Security could be based on user, device, and identity properties.
- The local OS could be specifically built and tuned for the capabilities of the device (power consumption, performance, user interface, etc.)
So what we end up with is a whole bunch of different "desktops." An iPad runs a local iOS desktop. A mobile phone runs an Android OS desktop. A laptop runs a Mac OS X desktop. A desktop runs a Microsoft Windows 7 desktop. A netbook runs a Chrome OS desktop.

So we move from a world where a "desktop" is just all of the aggregated apps, settings, and data, presented to whatever device the user is using at any given time.

Does this mean that Microsoft Windows is dead? Of course not!

First of all, there are millions of Windows apps in the world. And even if the world moves away from Microsoft Windows as the predominant device OS in the future, these Windows apps aren't going to disappear overnight. (And they're not going to disappear in the next five, ten, or twenty years either--look at how many mainframes are still out there!)
But what will happen is that we won’t need Windows running on the client device in order to access these Windows applications. Windows will evolve from a “local desktop OS” to an “application middleware” for delivering Windows apps to any device, including non-Windows devices. The Windows apps might run locally in an app VM, or they might be delivered from the datacenter via Remote Desktop RemoteApps, VDI, or Azure app clouds.

This also doesn’t mean that Windows as a desktop OS will disappear anytime soon. I’m sure that in 2020, you’ll still be able to walk into an electronics store and buy a laptop running Windows. But in that case, the fact that that laptop is running Windows is arbitrary. It runs Windows because the local laptop has to run SOMETHING, so it might as well be Windows if that’s what that OEM and that user want. But that’s “Windows the local device OS,” which is very different from “Windows the app platform for the enterprise.”
Of course that Windows laptop will be able to access Windows corporate apps, as will laptops running Chrome or Mac. And the organization will be able to deliver Windows apps to that Windows laptop, although from the corporation’s standpoint, they don’t give a hoot one way or the other what OS their users’ devices happen to run.

VMware Horizon App manager aims to take that one step further by helping to deliver any application (Windows, SaaS, whatever) to any device, even mobile phones. Horizon is one of a handful of products in a fledgling space that integrates your directory to the directories of other application and data sources so that you can manage, provision, and monitor application usage no matter where the application resides or what platform it runs on. What a world!