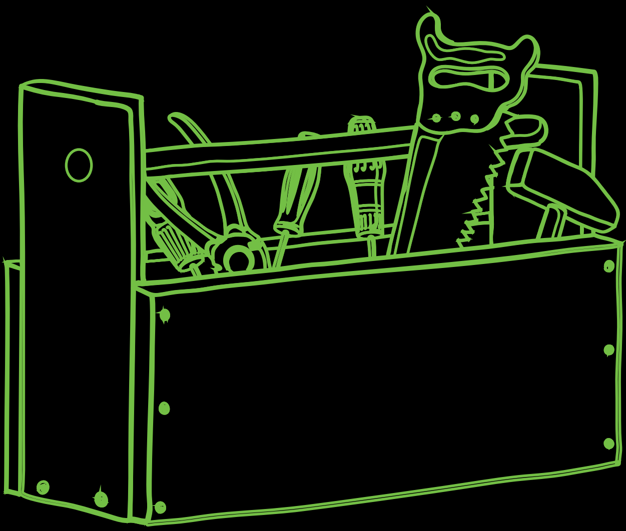
THE NEW VDI REALITY

How the biggest barriers to VDI adoption have finally been solved, and what this means for the future of the enterprise desktop



CHAPTER 14

The Future of the Desktop

Brian Madden

with Gabe Knuth & Jack Madden

It's human nature to envision the future only in terms of the present. Remember all the iPhone predictions from Apple fans that were published before it was first announced in 2007? They all looked like iPods or Nokia smart phones. Henry Ford was famously quoted as saying, "If I had asked people what they wanted, they would have said, 'a faster horse.'" This is why when you ask people to predict the future of the desktop, they describe things like web-based browser desktops that still have icons and wallpapers and Start buttons.

But after reading this entire book, you now understand that the desktop of the future isn't going to look like a Windows desktop in a browser. You know the desktop is a concept, not a concrete thing. So for us to answer the question, "What will the future desktop look like?" we have to figure out which desktop the person asking the question is talking about. Are they asking about the future of Windows or the future of the PC? Maybe they're asking about the future of devices with keyboards and mice? Or perhaps they want to know about the future of end-user computing in general?

Since it's possible that you'll run into people asking any of these questions, let's take a look at each of them one by one.

The Future of Windows

We've spent a good portion of this book discussing the future of the Windows OS and Windows applications. So if anyone ever asks you about the future of Windows, just give them your copy of this book. (Actually on second thought, tell them to buy their own!) Just keep in mind when discussing the future of Windows that there are actually two questions to answer:

- How will we deliver existing legacy Windows desktop applications?
- What will the Windows apps of the future look like?

Let's take a look at each of these.

The future of legacy Windows desktop applications

We recognize that it can be a bit reckless to throw around words like "legacy" to describe the millions of existing Windows desktops applications that are out there today. We certainly don't want to offend anyone, especially those who are in the midst of developing, buying, and deploying these kinds of Windows applications. That said, Microsoft has made it clear with the Windows 8 touch-mode apps, the Windows Store, and WinRT-based apps that the Windows native applications of the past twenty years are legacy.

Of course we all know that these legacy Windows desktop applications will continue to be developed, and we're sure that they'll continue to be used for decades to come. Notwithstanding everything we've written in this book so far, it's our belief that legacy Windows desktop applications will make a slow, yet inevitable, migration to the datacenter.

This move to the datacenter will take decades, and in fact, we don't advocate that you run out and try to move all your existing Windows desktop applications to the datacenter anytime soon. As we said previously, there are some great use cases for putting Windows desktop applications in the datacenter, but running them locally on client laptops and desktops is just too cheap, easy, and compelling today. So right now we only want to move them into the datacenter when it makes sense.

Over time, however, we'll see that legacy Windows desktop applications make up a smaller and smaller portion of our users' overall application set. This trend is visible already. Ten years ago, most of us had 100% Windows apps. Five years ago, we might have had an 80/20 Windows-to-web apps split. Now, it might be 50/50. If you play that out a few years, you'll see that we'll eventually get to the point where we only have a small handful of legacy Windows desktop applications. At that point, we'll have to ask ourselves, "Do we really want to support Windows and everything that goes with it on every client just to make a few legacy apps work?"

If you're dreading this inevitability, it's quite possible it won't be as bad as you think. For example, at BriForum Chicago 2010 (our own desktop virtualization conference), Atlantis Computing founder Chetan Ventakesh gave a brilliant talk where he explained why our Windows desktops and legacy applications are destined for the datacenter.

Chetan explained that Moore's Law is worthless when it comes to distributed desktop computing. Sure, it's great that we can get more processing for our money each year, but desktop computers are more or less stuck at the same price points they've been at for the past decade, and doubling the processing of a desktop doesn't change the computing model at all.

In the datacenter, however, we can apply this concept of "dematerialization," meaning physical objects are transformed into an abstract concept. Dematerialization of the desktop provides the liquidity whereby a desktop doesn't have to run within the boundaries of a single box. Chetan's not talking about flowing an entire monolithic desktop VM from one server host to another. Rather, he believes we'll be able to break up the memory, disk, data, CPU, and personalization so that each can run in the most performant and appropriate way.

Chetan proposed that the Windows desktop will grow to fill the boundaries of the datacenter, rather than the boundaries of a single computer. For example, by 2015:

- The rack is the new computer
- 10 Gigabit Ethernet is the new bus
- The hypervisor is the new kernel
- The software mainframe is the new OS

(Not bad for something presented in 2010!) In order to get this type of liquidity, the desktop can't run on a client—it's got to run in a datacenter. While the datacenter is an expensive and complex place to run Windows desktop applications today, Moore's law is making it more attractive every year. In 2010, we could only run 70 desktop VMs per physical server, which means we could fit 1,120 desktops in a rack. In 2013, we can run 150 desktops per server, or 2,400 in a single rack. Playing that forward, we'll be able to run 4,800 desktops in a rack by 2014, and 9,600 by 2016. Throughout that whole process, our per-desktop VM hardware cost drops from $400 to $150.

As desktop VM density increases, we'll see the boundaries of a VM break down. We already have some pretty amazing technologies for VMs in the datacenter, like Cisco UCS, block-level single-instance storage, memory-based disk caching, and the ability to boot live VMs with no storage. IOPS, once the killer of VDI, are now manageable. In the meantime, the amount of IOPS and CPU that a given version of Windows needs remains constant even as hardware gets faster. (Of course, Microsoft's desire to support that given version does not, so we'll probably still see hardware requirements inflate for new Windows versions over the years. But if you're just using Windows in the datacenter for legacy applications, you ought to be able to deliver them more cheaply each year.)

All these technological advancements mean that running Windows on VDI in the datacenter will be able to deliver a better experience than what's possible when running Windows on a client. In his 2010 talk, Chetan said, "Imagine that everything is instant. Apps open instantly. Docs open instantly. Everything is snappy and perfect. That's the experience that a dematerialized desktop running in a datacenter can deliver." At that point, the users can vote with their feet, so to speak. Combine that with the security, reliability, and falling costs, and Chetan believes that VDI is a no-brainer for the majority of use cases for Windows desktop applications.

Chetan closed his talk with this final thought: VDI is not just the sum composite of knee-jerk reactions to PC management, but rather it's a long-term transformational vector—the natural evolution of computing, and something that can't be ignored. (And we're seeing that now, in 2013. It's the New VDI Reality!)

Windows applications without Windows?

One axiom we've repeated throughout this book is that having even one single Windows application requires that we have Windows running somewhere. Some people wonder if that's actually true. They point out the open source project called Wine that attempts to re-create the Windows APIs and kernel calls in a software layer that can run on non-Windows operating systems. Wine is pretty amazing—you can literally run Windows EXEs on Mac or Linux OSes. The problem is that because Wine is reverse-engineered, it's always a few years behind in terms of what applications work. (At the time of this writing, Office 2010 doesn't yet work with Wine, and even Office 2007 has some major issues.) Because of this it's been easy to brush off Wine as a fun experiment, but not something that enterprises can trust.

Looking ahead, however, we have to wonder if that might change? If you believe that the biggest headaches in the future will be with legacy Windows desktop applications that can't be updated, you've got to think that at some point Wine will catch up. Sure, there are problems with Microsoft Office, but that's a huge suite, and there already are office suites for every platform—desktop users in the future aren't going to care about getting Microsoft Office from a Windows VM on their MacBook. So if the big concern is getting all these old proprietary Windows applications to run in the future, maybe Wine will work well.

On the other hand, running Wine still requires a full traditional OS. So if you use Wine to take Windows out of the picture, what do you really gain by replacing it with Mac or Linux? Are the Windows licenses that big of a deal?

Windows apps via HTML5?

Another concept we've covered quite a bit in this book is that it's possible to deliver Windows applications to users via HTML5. We focused on the various HTML5 clients for standard Windows applications running in the datacenter—things like VMware AppBlast or Ericom AccessNow. But there's another possibility. Companies like Framehawk are building solutions that let admins create HTML-based front ends for existing Windows desktop applications. The final result is apps that look and feel like native HTML5 apps, instead of the HTML5 Windows remoting clients that give users Windows-looking apps inside a browser.

Is this the way the last few Windows apps will be delivered from the datacenter in ten or twenty years? Perhaps.

Will Windows layering ever take off?

Throughout this book we've also touched a bit on a concept called "layering." Layering is the idea that you can slice Windows up into individual layers that are managed independently. You might have one layer for the base OS, another for corporate applications, and a third for user-installed applications.

Layering is not a product from a specific vendor. Instead it's more of a feature of desktop virtualization products or the description of what happens when you combine OS image management, application virtualization, and user personalization products.

The reason we mention layering in our discussion of the future is because layering is a hot topic right now. People initially believed layering would be necessary for VDI since they assumed that VDI could only work when many users shared a single master disk. (In those cases layering was thought to be the silver bullet that enabled users to have custom desktops based on a that common image.)

But advances in storage technologies have addressed the performance issues we initially thought only layering could solve. So now the conversation about layering has shifted to focus on leveraging it as a way to manage Windows desktops.

For example, some view layering as an application management solution. If you build a layer around each application, you can instantly enable and disable access to individual applications by turning a particular layer on or off. Another use is to support user-installed applications (UIAs)—the idea that users can install whatever they want into their own layer that is isolated from the base OS, (meaning IT can still refresh or patch the base layer without affecting the applications that the user has installed).

At the time of this writing, we don't know how this layering technology will fare and where exactly it will be the most popular. We know that the Windows OS and Windows desktop applications must be managed, and if doing so via layering makes more sense in virtual environments than traditional desktop management tools, that sounds great to us.

On the other hand, using layering strictly for user personality doesn't make as much sense to us. We spent this whole book writing about how users want their environment to work across platforms, so building up all these Windows-specific personality layers doesn't seem like it has much potential outside of Windows. We rather prefer the idea of creating the user personality as a separate thing that could ultimately be transferred across operating environments and devices.

The Future of the PC

The main shift of the PC over the past decade has been that ten years ago, the PC was the center of a user's world. It was the master copy that held the user's apps, data, and settings. It was everything. But nowadays, the PC is evolving to become just one of many consumption devices—in this case, in the form factor that allows for multiple displays and that has a keyboard and mouse.

This evolution wasn't based on any grand plan; rather, it was borne from necessity. Back when it was common for users to have one (and only one) computer, it was fine for that to be the master storage location for everything. But as we discussed in Chapter 10, smart phones, tablets, and Internet-based syncing means that the

PC is no longer the linchpin that held the master copies of everything.

Does this mean the PC died? Of course not. Sure, the role changed. Now, instead of using a PC because we have to, we only use a PC when we want a full keyboard or multiple huge displays.

This evolution will continue. PCs (as compared to tablets, smart phones, or thin clients) have certain characteristics that will still have value in the next decade—they can support multiple displays and run many applications can run at the same time. They provide huge amounts of processing power locally. They have full-size keyboards that enable people to type at hundreds of words per minute and they have precision mice and pointing devices. And of course in the Ultrabook form factor, PCs enable all of these capabilities to be packed up and taken anywhere.

So it's true that the PC will relinquish its position as the center of a user's universe, but it's not going anywhere anytime soon.

The Future of Devices with Keyboards and Mice

Perhaps this goes without saying, based on what we just covered with our thoughts about the future of the PC, but we don't believe devices with keyboards or mice are going away anytime soon, either. There are some who believe that speech recognition will replace keyboards, but we just don't see it. Have you ever tried to dictate a document instead of typing it? It doesn't work for most people. This has nothing to do with the quality of the speech recognition programs. The problem with dictation is that speech comes from a different area of your brain. Speech is very linear, where typing allows you to mentally jump around and visually construct sentences.

Even if you were good enough to speak your way through your typing, how's that going to work when other people are around? It's bad enough that we have to listen to everyone's phone calls near us—now we have to listen to their typing, too? No thanks!

Finally, even though humans are not good at multitasking, most of us type and take notes while we're on the phone. How would that work if we have to talk to our computers?

We believe that there is a place for speaking to your devices. (Asking your phone a question while you're driving is brilliant.) But just like tablets didn't replace PCs, speaking to your computer isn't going to replace a keyboard.

The Future of End-User Computing

Everything we've discussed so far in this entire book—the applications, the data, and the devices—could broadly be grouped into something called "end-user computing." By now you ought to be able to see how it's all going to come together.

Users will seek out applications. They'll want to use them via multiple devices with multiple form factors. The applications will increasingly store their configuration and data in ways that disambiguate them from their devices. (Start the email on the phone, finish it on the laptop.) Cloud-based storage of everything will become ubiquitous, regardless of whether it's public or private, user-based or company-based. All the while, IT will continue to lose control over what devices users can use and how they can access data. Whether the user selects and owns a device or the company does won't matter. IT will control access to corporate apps and corporate data, but beyond that, the users are on their own.

The desktop of tomorrow won't run Windows. But it also won't run the Mac OS. Or iOS. Or Android. The desktop of tomorrow is not a Chromebook or based on a browser. The desktop of the future is whatever device the user has in his or her hands at any moment. It will have the user's applications, data, settings, and personality. It's nowhere and everywhere at once.

And it's going to be here long before Windows is dead. So get on it!

—Brian, Gabe, & Jack. May 2013