

Snug As A Bug In A Rug. Perhaps...

Jeffrey R. Harrow
Principal Technologist, The Harrow Group

It's a warm, comfy feeling to sit down at your computer and work on your great new presentation, or on your proposal for your company's next great product. Perhaps you're working on an accounting spreadsheet that, while it lays bare some unhappy facts, might yet be "spun" in ways that change a potential disaster into a win in the market (all within the bounds of the GAAP, of course.)

One of the reasons that you can be so comfortable to carry on such work with your gently humming confidant is that, in most cases, your give-and-take *is* just between you and your CPU. Malware and spyware aside, you have reason to believe that you are having a private and intimate interaction with the data you're working on.

The problem is that you may become your own worst enemy. As we'll see, we're talking about approaching a *very* enticing but oh-so-slippery slope!

Step 1: The Way Things Were.



Let's look at how our computing experiences *used* to be.

You'd load an application onto your non-networked computer from a manufacturer's CD; let's say a word processing program. You'd fire it up, type to your heart's content, and periodically save the document to your hard disk. There it would sit, happy in its magnetic way, knowing that short of a physical break-in or subpoena your words were *yours*, relatively safe from prying eyes. Moreover, since you always backed up important files to a separate backup medium (writeable CD, external hard disk, even a (shudder) floppy), your data was as safe as you could make it.

Step 2: Enter The Network.

You know what's coming next. Along with the vast benefits that come from plugging in that Ethernet cable, you've also made it much easier for "black hats" to reach in and touch your data. They might be nefarious characters out to steal personal data or company secrets, or they might be trying to seduce your computer into joining an army of "bots" that can be used for a number of purposes that you probably don't support. Or for other unsavory tasks.

But you (or your network administrator) have taken precautions: You have a firewall, anti-virus software, anti-malware programs, and anti-spyware protections that together, if kept updated, give you a fighting chance of retaining the privacy of your data -- just like in the "good 'ol" pre-network days.

But you do have to keep up with the (software) Jones.

Step 3: Do It Yourself.

There's no way around it -- maintaining software (security tools and applications alike) is a pain. There are patches that fix bugs. There are patches that add new functionality. There are "minor upgrades" that often require you to download and install them. And there are, periodically, entirely upgraded versions that have been known to "break" the function that you use the most (but don't worry - that will be fixed in the next patch.)

Keeping up with software can also be expensive -- there are all the person-hours that you or your IT team have to dedicate to the care and feeding of your applications (which leads one to wonder "who is working for whom," but that's another discussion.) Then there are the upgrade fees often associated with a major release of your applications. Not to mention the additional time that may be required to troubleshoot new after-upgrade problems that might show up.

It almost makes you want to go back to pen and paper! (Not.)

So it's no surprise that people have been looking for the proverbial "better way," and a growing number of folks see that in "Web Services."

Step 4: Do *Nothing* Yourself!

There are lots of [definitions](#) of Web Services, and many people view various implementations in different ways. Taking a few liberties to put a Web Service in

context, it is basically an application (program) that you use, but that *does not run on your computer*.

(By the way, as we get more of a sense of what Web Services are, you may be thinking that you're a "computing traditionalist" and don't or won't use these new Web Services. If so, you'll shortly be in for a surprise.)

Typically, a Web Service application runs on a server somewhere out on the Internet (or perhaps within a large company's intranet). You connect to the Web Service using your Web browser or sometimes using a small specialized access application. Essentially, your computer provides the interface, but the Web Service's remote computer provides the data and the computational "heavy lifting." Assuming that your network is reasonably responsive you don't really know (nor care) that it's someone else's computer that is crunching your spreadsheet, manipulating your slides, or storing your word processing document.

And it's simpler -- you (or your IT department) didn't have to load all of those applications in the first place (they live on the Web Service's computer, wherever it might be); you don't have to worry about keeping the application patched (the Web Service's IT staff does that once at its central location and, magically, the next time you access their (your) application over the network you're using the patched version. This also applies to minor and major upgrades to the applications. Using Web Services for typical office applications may also prove to be less expensive than installing them onto every computer in your business; you might only pay for users who actually use those applications, or pay by the minute, etc.

Not a bad thing at all. And Web Services can get even better. While you *can* still choose to keep the fruits of your labor (your documents, spreadsheets, etc.) on your hard drive, there's a pretty good enticement for you to store them on the Web Service's server -- your documents are then available to you from wherever you log in, without your having to carry them around on a USB pocket drive or other storage medium. Additionally, you're always working on the latest version of your document, not a version that you forgot to update on your USB drive that doesn't contain the important changes that you made this morning.

Speaking of your documents, a Web Service's offering just might include their backing up one or more versions of your documents in one or more backup locations (depending on your documents' importance to you.)

A few examples (no recommendation implied) that you might explore to get a sense of how Web Services actually work include: Microsoft's [OfficeLive](#); [SimDesk](#); [ThinkFree](#); [NetCubicle](#); and the recently released beta of [Google Spreadsheets](#).

Aspects of Web Services certainly do qualify as goodness and light. But...

Enticing, But Not All Sweetness & Light.

It would be so nice if Web Services were the answer to all of our computing needs, and depending on each situation they might be. But as with anything else related to computing, there are some issues that we should all be aware of so that we can make informed choices based, in part, on the sensitivity and criticality of our work.

One issue is that communicating all of your data to a Web Service provider may leave the data stream between your computer and the Web Service's computer open to prying electronic eyes. Is the data stream effectively and always encrypted?

How about surreptitious access to your data once it's safe on the Web Service's computer -- do they vet their employees? Do they keep nasty types of computer vermin at bay? Is their location physically secure? How about environmentally secure? Backup power? Do they mirror their operations at a backup site in case of disaster? Do they mirror your data in real time (preferably off-site) so that you won't lose your work if something goes bump in the night? Do they have "hot failover" in case the computer that happens to be providing your application fails at 2AM while you're finishing up the presentation for your Board of Director's meeting at 7AM?

Then there's the issue of your working on that critical project when your Internet connection goes down, or that you might need to do the work on an airplane or anywhere else where you can't reach out and touch the Internet -- are standalone applications maintained on your computer that can work with the Web Service's file formats? If so, does the Web Service application automatically store an up-to-the-second "emergency" copy of every document you're working on directly on your computer to support planned or unplanned standalone work? What plans are in place to safeguard your information if the Web Service company goes out of business? And of course there are many more concerns, and benefits, when considering using Web Services.

Overall, the most important question is, after taking everything in *your* particular situation into consideration, can you trust at least some elements of your businesses' core information to another business?

Web Services 'R You -- Already!

Surprisingly, Web Services are not something to consider "mañana," since they *already* go far beyond "office apps!" In fact, you probably already use more than a few Web Services. Ebay is a Web Service (all the work is done on their computers while your computer just provides the interface.) In the same way, Amazon.com fits the bill. As well as the online shopping sites that you may frequent. [Google Maps](#) and [Microsoft MapPoint](#) and [Windows Live Local](#) are all Web Services. As is the magnificent [Google Earth](#). Similarly, there are many online phone books. Many newspapers to read as you drink your coffee. There's buying something with PayPal. Indeed, Web Services are sneaking under the radar and are *already* becoming pervasive.

In fact, just about every application that you access through a Web browser is a form of Web Service!

Yes, some are more sophisticated than others. Searching an online phonebook does simple searches against a database. But programs such as [Google Earth](#) (try it if you haven't!) conduct a complex and constant dialog between your computer and its geographic database to constantly clarify areas that you're looking at, as well as to download areas that you *might next* look at (based on where you're looking now); it happens in the background so those new sections of the map are ready if and when you are.

Of course these seemingly trivial activities of browsing phone numbers and annotating maps might not seem to deserve the "information safety" efforts that we discussed in relation to Web Services-based office applications -- until you begin using that phone and map information to depict your business data!

Psst - Have I Got A Web Service For You!

We now know that Web Services are conceptually simple and are easy to understand. They're also very much here, and are on the road towards becoming real contenders for computing tasks that we've traditionally conducted within the sanctity of our PCs. This is not a 'bad' thing. But like so many issues at the boundaries between business, society, law, and rapidly moving technologies, these are changes that deserve careful thought, and perhaps careful legislation, to protect new users.

The only change that we can be sure of is that "change happens." And it will continue to happen ever-faster.



www.futurebrief.com

Don't Blink!

This essay is original and was specifically prepared for publication at Future Brief. A brief biography of Jeff Harrow can be found at our main [Commentary](#) page. Other essays written by Jeff Harrow can be found at his [web site](#). Jeff receives e-mail at jeff@theharrowgroup.com. Other websites are welcome to link to this essay, with proper credit given to Future Brief and Mr. Harrow. This page will remain posted on the Internet indefinitely at this web address to provide a stable page for those linking to it.

Please feel free to share the PDF with others who may be interested. To hear about future **Commentary** essays, take a few seconds to read about [Daily Brief](#), one of the "briefest" Internet updates offered anywhere.

© 2006, Jeffrey Harrow, all rights reserved.