

Browser-based vs. Client-Server-based ELNs – Which is right for you?



By Wolfgang Rumpf, Ph.D.

There are many Electronic Lab Notebooks on the market, and picking the right one for you can be confusing. One of the issues that I often run across is that of “is it cross-platform” (meaning will it run on Windows as well as on Mac OS X). And while this is an interesting topic on its own, it made me think more of what cross-platform really means.

In the old days, cross-platform meant compiling code so that an application would run natively on more than one operating system. And in more recent times, developers started to use Java™ to create applications that were cross-platform in that they would run natively in a *virtual machine*, and virtual machines were available for many (if not all) platforms.

To many ELN vendors, however, if the software runs within a web browser, it is cross-platform. What does that really mean? It means that maybe you have a document manage system (like Microsoft Sharepoint™), and perhaps other functionality, that lives on a server but that users access via their web browser (Internet Explorer, Firefox, Safari). Yes, users on both Windows and Mac (and even the various flavors of Linux) can access this information and generate content. But what does this really mean for the users?

Any application is at the mercy of the technology it is built upon. Browser-based applications are at the mercy of the browsers they run within - and therein lies the problem. Not all browsers are equal, nor do they all render the exact same content in the exact same way. And not only that, but browsers are generally designed to retrieve and view content, not to serve as authoring platforms.

The end result? Browser-based ELNs - while cross-platform, and functional - often provide the user with a less-than-optimal experience when compared to their “heavy-weight” application-based cousins. Users may not experience the same sort of visual experience when viewing information, and content creation may not be open and free-form, relying instead on uploading of existing files or typing text entries into web forms.

Another disadvantage - the heavy lifting (any converting, storing, analysis, processing, etc.) is done on the server, meaning that responsiveness and scalability will be affected.

The advantage, of course, is that browser-based are generally significantly *cheaper* than their application-based cousins. And while I genuinely appreciate the limitations of fundings we’re all facing nowadays, quite honestly you get what you pay for.

So then what are the advantages (and disadvantages) of Client-Server Application-based ELNs?

For one, since these are actual applications, the visual interface is very tightly controlled - and thus consistent - for all users. And the application can be much more dynamic, incorporating live data components that let you view not just a static representation of your data, but the real live data - something you can interact with and manipulate in real time.

Another advantage - the client on the local computer takes some of the heavy lifting burden away from the server, which means that these systems can be more scaleable. Of course this does mean that the client machine needs to be somewhat more robust - a (minor) disadvantage.

Another disadvantage of application-based ELNs is that they frequently are *not* written to be cross platform, because it can take significant effort to transfer the code to another platform. And since the Mac OS X and Linux platforms represent a small (significant, but still small) proportion of the potential ELN users, most vendors end up *not* supporting anything other than Windows. Or they end up writing a web browser-based application. In fact there are only two vendor I know that writes an application-based ELN that runs on both Windows and Mac OS X platforms, one of them being [Systat Software Inc.](#) (if you know of another, please let me know and I'll update this entry).

The other disadvantage to application-based ELNs? Cost. Typically anywhere from 50% to 150% greater in price, although your mileage may vary, and IMHO you definitely do get what you pay for.

My ELN recommendation? I refuse to make one. I'm biased, you see, and quite honestly what works well for one person may not work well for the other. I do strongly recommend three things:

1. Take a look at several different vendors, and test-driving any system you think may be right for you.
2. Don't take the vendor's word for *anything* - be sure you talk to existing users of all the systems you look at and see what *they* think of the application.
3. If cross-platform functionality matters, don't buy an ELN from a vendor that tells you "Oh, we're working on that, we'll have Mac/Linux support any day now". One vendor I've talked to has been saying that for a *long* time (no, I'm not going to name names - you know who you are). Buy an ELN that works for you *today*, not one that promises you the moon *tomorrow*.