STARLIMS®

WHITE PAPER

# Web-based vs. Web-enabled LIMS: What is the Difference?

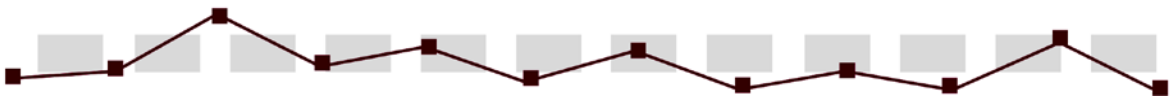
October, 2007

Simon Wood, PhD

Executive Director, Marketing and Education

TABLE OF CONTENTS





# HISTORICAL BACKGROUND

Commercial Laboratory Information Management Systems (LIMS) have been available on the market for about 20 years. Given the phenomenal rate of change within the IT industry over that time, it is not surprising that there has been an equally significant change in the technology underlying commercial LIMS. Indeed the LIMS industry over the past 20 years has been characterized by arguments and debates over the relative advantages of the available technology.

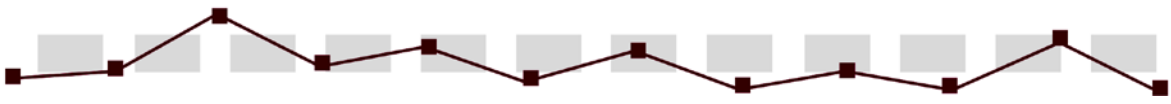
Initially LIMS were available on mainframe or mini computers such as Dec Vax systems, with users accessing the system through character-based terminals. As PCs became more and more ubiquitous, the technology shifted to Client/Server-based systems running over Wide Area or Local Area Networks (WAN/LAN). As a simple definition of Client/Server Systems the Client (i.e. the PC on the user's desk) runs some, or all, of the application while the server runs the rest of the application, hosts the database in which the data is stored and process requests for storing and retrieving data. Until the advent of the Web and the growth of Web-based Technology, Client/Server systems were the dominant force in LIMS. However, even at this time there was argument as to how much of the application should be on the client and how much should be on the server, the so-called thin client/ thick client debate.

When the PCs available to users were relatively short of computing power (mainly due to the cost of memory, processing power and storage) the client side of the application was limited in its capability and would be responsible for relatively little processing i.e. they were thin clients. For example they might just be responsible for managing the graphical user interface. Massive improvements in the performance of available PCs (caused by a reduction in cost of processing power, memory and storage) meant that much more powerful machines could be made available to users on their desk tops. These machines could do more and more processing independently of the server, so more and more of the application could be run on the client i.e. they became thick clients. Various manufacturers adopted either thin or thick client architecture but the overall move was towards more and more of the application running on the client.

With both thin and thick clients there is always application code that needs to be loaded onto the users' machines for the application to run. The application must be installed on every machine that needs to run the application and when updates become available they must also be installed on each machine. For organizations with large numbers of LIMS users this can become a logistical nightmare, with the constant concern that users could be using different versions of the same product if the upgrade is not properly managed. Validation of the installs also becomes a concern, especially within the regulated industries. The introduction of code management systems and automatic deployment mechanisms helps overcome some of these issues to a certain extent, but the issue of multiple machines each running the application code and the need to manage these machines remains (including ensuring they are powerful enough to run the latest version of the application). Of course, if a user's PC should go wrong they will only be able to continue working

when they can obtain or use another machine with the application installed—an important issue when dealing with business-critical systems and functions such as those managed by LIMS.

The adoption of emulation software such as Microsoft's Window Terminal Services and Citrix Metaframe can help alleviate some of these issues. Implementing emulation software of this type means that the application runs, and all processing is carried out, entirely on the server, but the application and interface are presented to the user as if they were running on the PC. This technology makes it easier to roll out the application to users, but of course the emulation software itself needs to be maintained and supported, and if users see problems with the application it can be difficult to trace these to the application itself or to the way it works within the emulation software.



# WEB-BASED LIMS

The adoption of the Web as a serious tool for business and the subsequent demand for systems that run within a Web environment has triggered another change in LIMS technology with users wanting Web-based systems. LIMS suppliers have responded in a number of ways and the way they have responded, and how they characterize their offerings, can cause possible confusion among potential users. Typically systems may be described as Web-based or Web-enabled, and confusion can arise between these terms, especially as they are sometimes used interchangeably, even though they are technically very different. It is important that users understand the differences so that they can identify which systems are truly Web-based and which are Web-enabled when considering the options available.

Web-based systems are characterized as having been developed specifically to run within the Web environment. The 'client' is an internet browser and is responsible only for displaying the user interface; none of the application runs on the client side and none of the processing is done there either.

Early Web-based systems were implemented using HTML and were characterized by having simple user interfaces and limited functionality, although they did allow efficient navigation of the Web and expanded accessibility to the applications. However, the limitations of HTML meant, for example, that it was not possible to implement complex simultaneous processes. Within LIMS such process are commonly required. For example on result entry the system may need to trigger a calculation to run and a change in status, as well as validation of results against specification and updating of the screen display.

Developments in programmable eXtensible Markup Language (XML) and SOAP-based Web services now allow developers to provide rich user interfaces and to implement complex functionality. Static HTML pages have been replaced by rich interfaces created with XML form engines, meaning that Web-based systems can now have User Interfaces equivalent to Client/Server based systems in terms of usability, functionality and familiarity. Communication between the Web browser (client) and server is achieved using the standard Web transfer protocol (HTTP) or optionally the Secure HTTP (HTTPS) and Intranets or Extranets with firewalls can be used.

However, by fully leveraging the .NET Framework within the web browser, true Web-based systems can provide much more than just a good looking user screen. The browser becomes a powerful web based 'client' that incorporates the full power of the .NET Framework. Interfaces to physical devices as well as other local or enterprise IT systems can be developed, implemented and managed within the Web environment. This is of particular importance within the laboratory where LIMS may need to interface with many different types of device and software systems. For example there may a need to make physical connections with devices such as cameras, balances or other simple instruments, for the capture and storage of data via physical USB or RS232 cables. The use of the .NET framework technology allows fully Web-based systems to manage these

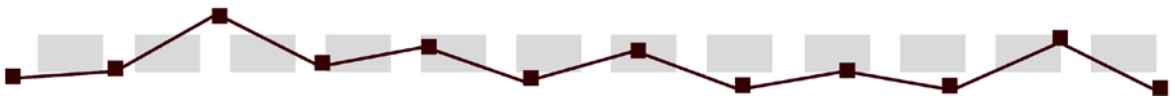
interfaces from the web browser as effectively and efficiently as any Client /Server system. The same is true for software systems where LIMS may need to interface with local laboratory based instrument software systems such as Chromatography Data Systems (CDS) or Mass Spec Data Systems, site based systems such as SCADA and enterprise wide systems such as ERP, or Scientific Data Management Systems (SDMS). A truly Web-based system will allow for the full integration of these systems all managed from within the Web browser; in effect the Web browser becomes the interface.

True Web-based Systems allow all system functionality including management and development functionality to be run from the browser; indeed by its very nature a true Web-based system can only run the functionality through the browser as there is no client side application code.

There are many advantages of adopting the Web-based approach. Based on what has been said before perhaps the most obvious is that there are no issues with maintaining client side software as the application runs entirely on the server. This also means that performance is dependent only on the servers available and is therefore more manageable. Both of these mean that management of the system is simplified. As Web-based systems have no client side software installed the potential for the application interfering with other applications is removed, for example conflicts that can arise if applications require different versions of the same shared components.

Access to the system is simplified as well, especially for remote or geographically separated users. Remote users only require an Internet connection in order to use the system. This is obviously an advantage for home based users or for users who spend periods of time away from their office or laboratory. However, Web-based systems also provide the ability to easily extend the system to customers, or other third parties, who may require access. Provided that the system allows for user or role based access definitions, the nature of Web-based systems mean that these external users can have access to as much or as little of the system as required, without the requirement to create additional specific functionality. This is because they are using exactly the same system as internal users. Of course this means that it is simple to allow customers to create their own testing requests, review results and query their information in the same way that internal users would. It also means that if testing is outsourced to third-party laboratories or testing facilities these organizations can enter results directly into the LIMS, removing the need for them to submit result reports to the laboratory for input. Again as zero client side software is required for these users, there is no system management burden imposed when upgrades and new releases become available. If external users require access to additional functionality within the system, this can be achieved by simply changing their role or access definitions.

Web-based systems that support and use Web Services also enable full enterprise integration to other systems that also support Web Services. Within the laboratory there is the need for seamless integration between many different systems including LIMS, Chromatography Data Systems (CDS), and Mass Spec Data Systems (MSDS). However, integration to other enterprise systems can be equally important i.e. Enterprise Resource Planning (ERP) Systems, Manufacturing Resource Planning (MRP) Systems, Document



Management systems, Data Warehouses and even Human Resource (HR) systems. The adoption and use of Web Services can greatly simplify both the development and maintenance of these enterprise links.

## WEB-ENABLED LIMS

Web-enabled is a somewhat undefined term but can be summed up as an existing Client/Server system that has had some features developed that allow it to run all, or some, of its functionality within a Web Browser. This can take two main forms, with a third possibility of employing Web-based deployment technology. First, there is the development of Web-based front ends that provide a portal like interface to the system for specific, but usually limited, functionality. For example, options for users to view results or sample status or to pre-log testing requests. These options are often simple and provide an unsophisticated user experience and limited functionality. As they have to be developed specifically for the application, every time an external user requires new functionality this has to be created; they cannot use similar functionality that may be available to non Web-based users in the Client/Server system itself.

Another approach adopted for Web-enabling LIMS allows Web connectivity of the system by replacing direct client to database connection e.g. open database connectivity (ODBC) with Web Services protocols e.g. Simple Object Access Protocol (SOAP), therefore making remote access to databases possible. While this approach allows accessibility to the central database from multiple diverse sites over the Internet and provides the user with a familiar user interface, it still requires the installation, update and maintenance of the client software on every users PC and therefore limits system accessibility to only those PCs with the client software installed.

Finally, some systems that are described as Web-enabled actually just use Web-based deployment strategies to deploy Client/Server systems. Products such as Microsoft's ClickOnce deployment technology can alleviate the maintenance burdens of a Client/Server system by employing a Web-based application deployment and update strategy that simplifies the installation process. There may be a number of deployment strategies available, but in essence users click on a specific link or icon and the client application is downloaded and installed to their machine. In addition, updates can be checked for and downloaded automatically. Microsoft's ClickOnce technology also installs the client application on the user's machine into a separate 'sandbox' or self-contained protected area which helps to minimize the possible impact that the client application may have on the rest of the client system and other applications that are running i.e. version conflicts between shared components. However, these systems are still Client/Server based systems with the client application running on the user's machine.

## SUMMARY

There are several important distinctions between Web-enabled and Web-based LIMS. It is important to understand the differences between the two on a technological and functional level and to understand which model a particular LIMS or LIMS supplier has adopted. Making this differentiation can help users to make the correct choice of system that meets their Web strategy. Web-based systems are those that have been specifically developed to run in the Web environment; they are characterized by having zero client side install and all functionality—including management type functionality—running within the browser. While early Web-based systems may have been unsophisticated, especially in terms of the user interface, adoption of the latest Web technology ensures that these systems are as sophisticated and functionally rich as Client/Server based systems.

Web-enabled systems are Client/Server based systems that have had some, or all, of their functionality adapted to run within the Web environment. However, they still require client-side software to be downloaded, with the system management and possible system conflict issues that this involves. If the system has functionality that is not available from within the Web browser or if functionality is available both from within a Web browser and outside of the browser, then the system may be Web-enabled, but is not truly Web-based.