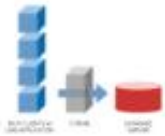


## Selecting the Right LIMS

### Critiquing technological strengths and limitations

Keith M. O'Leary

For many organizations, their Laboratory Information Management System (LIMS) is a mission-critical component of their overall corporate Information System (IS), providing integral connections to these companies' laboratory data, instruments, analysis and reports.



click to enlarge

Figure 1

Today, LIMS are in the fourth generation of evolution, and many organizations are faced with the need to upgrade, supplement or modify their LIMS to ensure optimum interoperability with the overall corporate IS infrastructure. Now, the proliferation of choices and terminology describing the solutions make selecting the best LIMS architecture for an organization more critical than ever to ensure successful management of the LIMS implementation and maintenance.

The task of selecting the best LIMS for an organization becomes even more complicated because many of the terms used to describe very different LIMS solutions are used interchangeably. This article will provide clarification of the terms as well as the capabilities, strengths and limitations of each type of solution.

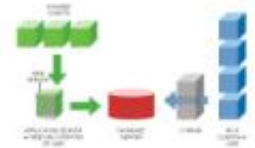
### The LIMS evolution to current web functionality<sup>1</sup>

LIMS connect the analytical instruments in the lab to one or more workstations or personal computers (PC). These instruments — such as chromatographs — collect sample data. An instrument interface forwards the data to a PC, or further to a server, where the data is organized into meaningful information. This information is then stored to be mined by the lab and other departments and/or sorted and organized into various report formats based upon the type of report required. Full-featured LIMS will manage the various laboratory data from sample login to reporting the results, and will address the needs of research and development to quality management laboratories.

The first commercial LIMS were introduced and developed in the 1980's by analytical instrument manufacturers. These first generation LIMS placed laboratory functions onto a single centralized computer, providing greater lab productivity and functionality as well as the first automated reporting capabilities. These were quickly followed by second generation LIMS which utilized third-party commercial relational databases (RDB) to provide application-specific solutions. Most relied on minicomputers, such as Digital VAX, but then, PC-based solutions soon emerged.

The increase in computer processing speed, the enhancements in third-party software capabilities, and, the reduction in PC, workstation and minicomputer costs paralleled the introduction of commercial LIMS. These advantages drove a migration away from proprietary commercial systems toward an open systems approach that emphasized user-configurability, rather than customization by the vendor.

The third generation LIMS, emerged in the 1990's, combined the PC's easy-to-use interface and standardized desktop tools with the power and security of minicomputer servers in a client/server configuration. Its architecture split the data processing between a series of clients and a database server that runs all, or part of, the relational database management system. By the mid-1990s, fourth generation LIMS decentralized the client/server architecture further, thereby optimizing resource sharing and network throughput by enabling processing to be performed anywhere on the network. When the Internet took off in 1996, the first web-enabled LIMS were introduced, followed by web-based and thin-client solutions.



click to enlarge

Most of these thin-client solutions were developed in Java, and web-based systems on Microsoft's .NET platform. Some of these web-based systems leverage XML, or eXtensible Markup Language, to transmit data between traditional client/server architectures and the .NET framework to offer a web-based application.

Figure 2

### Consider the client side options

Understanding the differences between thick-client, web-enabled, web-based and thin-client LIMS is challenging because the terms are often used interchangeably, adding to the confusion and making it difficult to develop a well-informed decision when it comes to choosing the "right" LIMS. Because these terms actually apply to very different types of architectures, a thorough understanding of the pros and cons of each is essential when evaluating functionality and the impact on implementation and deployment.

Thick-client is used to describe an application designed to run in a client/server environment, meaning that a portion of the software resides on the server and the other portion resides on the user's workstation. Clients are essentially the PCs or workstations. Also known as a fat client, it is a client that performs the bulk of any data processing operations itself and relies on the server it is associated with primarily for data storage. A thick-client has the LIMS installed on the PC hard drive. The thick-client connects to the database server, but the processing is done on the client side (See Figure 1). Each time the system administrator changes the configuration or customizes the application, those changes must take place at the client level, i.e., the modifications must be propagated to each individual workstation. In order to provide the interconnectivity of each of the end-user locations, thick-client LIMS often rely on cross-platform programs such as Citrix.

**Thick-Client.** The LIMS resides on the user's PC or workstation.

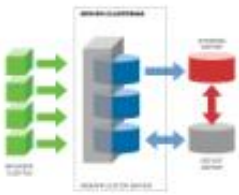
Strengths:

- Rich functionality can be made available utilizing the PCs local print, storage and multimedia capabilities.
- Real-time connection to the server is not required to perform functions.
- Processing speed can appear "faster" because much of the processing takes place on the PC and fewer trips to the server are required.
- Security is somewhat enhanced as only users with the client application installed on their PC can access server side information.

Limitations:

- More expensive end-user "rich" or "fat" clients are needed, since these clients perform the bulk of data processing, increasing hardware costs.
- Cross-platform programs are needed to support an enterprise deployment increasing initial and ongoing LIMS maintenance expenses.
- System administrators are forced to make changes to the LIMS configuration or customization on each end-user thick-client increasing ongoing LIMS support and maintenance expenses.
- No functionality available to users through a web browser.

Web-enabled is used to describe the add-on web browser component of an application designed to run in a client/server environment. The web-enabled portion of the application may allow access to data from a web browser, but the user is limited to the product functionality that is available on the web portion of the system (Figure 2). To access and utilize the full functionality of the application, users must have the local client installed on their desktops. In order to provide the interconnectivity of each of these end-user locations, Web-enabled LIMS also often rely on cross-platform programs such as Citrix. Similarly, modifications must all be propagated to each thick-client.



**Web-enabled.** Most of the LIMS resides on the user's PC or workstation, but add-on portions run through the web browser.

click to enlarge

Strengths:

- Enables both local access and server side access to stored data.
- Functionality can appear "faster" as some processing is still taking place on the PC or workstation.

Figure 3

- Combines the best of the thick-client functionality with the benefits of a familiar Web browser interface.

Limitations:

- Limits functionality available to users through the web browser.
- More expensive end-user "rich" or "fat" clients are needed, since most users require these clients to perform the bulk of data processing, increasing hardware costs.
- Cross-platform programs are needed to support most users in an enterprise deployment increasing initial and ongoing LIMS maintenance expenses.
- System administrators are forced to make changes to the LIMS configuration or customization on most end-user thick-clients increasing ongoing LIMS support and maintenance expenses.

An application that is thin-client or web-based offers end-users full application functionality from a browser. A thin-client does not have significant hard drive or memory requirement, as it does not store or process data. The LIMS resides on the application server(s) while the thin-client simply presents the screen display and allows users to use the keyboard and mouse to interact with the application server. With a thin-client application, any software configuration changes or customizations made on the server are immediately universally available to all the end-users. There is no need to propagate the changes to the end-user client, or leverage regional servers with cross-platform programs. This happens because the software resides on a central server and each time a user accesses the software, they are accessing the same current version from the web.

The distinction between web-based and thin-client solutions is the amount of software residing on the client. A true thin-client LIMS is an application written specifically to run on the web, with zero-footprint on the end-user client. It only requires a browser or "dumb" client terminal to run the application residing on the server. No downloads, applets or other programs exist on the end-user client, other than the browser (Figure 3).

**Thin-Client.** Provides full LIMS functionality through the web browser with "zero footprint" and no processing on the end-user's client.

#### Strengths:

- Less expensive end-user "dumb" clients can be used, since no data processing occurs on the client, reducing hardware costs.
- Configuration changes or customization are made on the application server, rather than on each client, reducing ongoing LIMS support and maintenance expenses.
- No need to rely on cross-platform programs reducing initial and ongoing software costs.
- No need to update any .NET framework reducing ongoing LIMS maintenance expenses and removing reliance on Microsoft supported browsers and application servers.

#### Limitations:

- Requires real-time access to the server for client functionality.
- High throughput capacity required of the network infrastructure increasing some of the inherent hardware and software costs.
- Some latency can be experienced due to the round trips required to and from the server for each action.
- Somewhat reduced level of functional interactivity and a slightly greater number "clicks" when compared to thick-clients.

Web-based LIMS built on Microsoft's .NET framework typically transmit data between the .NET framework and traditional client/server architectures to offer a web-based application. In such systems, the thick-client is somewhat hidden because it runs in a browser. However, there is more running on the client than just the browser. A .NET application that runs on the .NET framework installed on the end-user client is required to render the browser pages from the thick-client server. In essence, the browser is not showing HTML, but instead, is showing a thick-client-server-like application. In addition, these applications also rely on the local client for computer processing, and therefore require a rich, rather than, dumb end-user client. (Figure 4).

**Web-Based.** Provides full LIMS functionality through the web browser, but uses the .NET framework to download and run the LIMS application on the end-user's thick-client.

#### Strengths:

- Full LIMS functionality via a familiar Web browser interface.
- Some enhanced processing power due to the use of the locally available .NET framework.
- Some greater functional capabilities due to locally available .NET programs and applets.

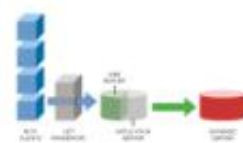
#### Limitations:

- More expensive end-user "rich" or "fat" clients are needed, since these clients share the bulk of data processing, increasing hardware costs.
- Requires use of Microsoft Internet Explorer with .NET and Microsoft supported application servers.
- System administrators are forced to update the .NET framework to make changes to the LIMS configuration or customization on end-user thick clients increasing ongoing LIMS maintenance expenses.
- May limit the use of certain PDA, tablet PCs and/or hand-held devices.

#### Trade-offs of each solution

Integrating a LIMS into the overall IS should take into consideration the company's current infrastructure, long-term plans for the IS, and investment in hardware and human resources to manage the IS. This decision in turn affects the cost of clients and servers, third party software needed to connect these LIMS clients, and the resources needed to deploy and upgrade the LIMS. Furthermore, it affects the robustness and security of the application as a whole, and the flexibility of the system to later modification. Depending on the answer and the trade-offs to be made, the organization may decide to use either a thin-client, web-based, web-enabled or a thick-client solution.

**Network Bandwidth.** Thick-clients and web-enabled solutions typically require less network bandwidth. Moreover, because thick-clients themselves do much of the application processing, they do not require an application server for processing. However, it is important to remember that the database will likely reside at the server level, so communication to and from each thick-client is still required. Additionally, regional hardware and software is required to operate the cross-platform programs, such as Citrix that enable such networked communication. Web-enabled and some web-based systems offer a hybrid to this approach, depending on where the functionality resides (i.e., on the client or on the server).



click to enlarge

**Redundancy.** Thick-clients provide some level of redundancy. In a simple LIMS solution that uses a single thick-client, if the server/network goes down, the client can still collect sample data and hold it on the PC's hard drive until recovery and data can be forwarded to the appropriate server. In a thin-client application, redundancy is achieved through clustered application servers that provide load balancing and fail-over, a process that routes client requests to servers within the cluster. If one or more servers fail, client requests are automatically routed to other servers within the cluster so there is no break in service.

Figure 4

**Performance.** Thick-clients tend to have advantages in multimedia-rich applications that would be bandwidth intensive if fully served. For example, thick-clients are well suited for chemical drawing and molecular modeling programs which require a significant amount of computing power. With that said, new technologies, such as Ajax (Asynchronous JavaScript and XML), are offering more dynamic user experiences than previously available over the web. Moreover, thin-client LIMS can be easily scaled by adding servers to the cluster.

**Access.** With a thin-client LIMS solution, users can access the LIMS from virtually any Internet-ready device. All of the lab's applications and data are maintained centrally, thus allowing any number of people to share them in a secure way by simply plugging in a thin-client browser. Thin-clients also enable connectivity by critical users external to the lab, such as executives,

customers and partners. Moreover, thin-client solutions can leverage a "dumb" client or portable device, as opposed to some web-based systems that require a client to leverage the .NET framework to communicate with the server. Thick-client systems pose the most difficulty in enabling enterprise access. Such systems must be installed at each end-user client, and have no remote access capabilities. Web-enabled systems offer some remote access, but only to the specific, limited functionality available via the browser.

**Security.** Thick-clients provide standard network based security as well as a slight security benefit from outside attack in that the client software has to be resident on the end-user's PC or workstation in order to access the server side data. This prevents non-client access or browser based external attacks on the server side data. With thin-clients, all the applications and data are maintained on a central server. For leading thin-client LIMS, access is role-based and password protected for security, complying to 21 CFR Part 11. Disabling a login account disables access to all company information. No application data ever resides on the client, hence files are prevented from being transferred to local hard drives and memory sticks.

**Ease of Use.** Thick-clients can provide a greater level of interactivity through the use of proprietary functionality that takes advantage of locally stored data, procedures and multimedia capabilities. However, it comes with the cost to locally store data and programs, the complexity of version management for the end user community, and the costs to train users on the proprietary functionality. The thin-client browser interface has the familiar web-site look and feel. Thus its use is intuitive, which keeps initial and ongoing training costs low. Because end-user adoption is easier, the LIMS can be integrated with the organization's workflows quicker. This enables organizations to leverage their investment and reap the benefits of enhanced productivity sooner.

**Hardware, Software and IT Costs.** Thin-client solutions reduce the cost per user and IT manpower requirements while delivering a solution that is easily upgradeable and expandable (without interrupting business). Software updates are done once from a central server and are immediately available to all; desktop upgrades are not required. A thin-client LIMS enables IT to deploy, maintain and support only one central LIMS, thereby increasing return on investment (ROI). Thick-client and web-enabled systems require IT to upgrade all end-user thick-clients around the world, along with each of the regional servers and cross-platform programs residing on them. Although some web-based systems achieve some of the benefits of thin-client systems, there are added costs to maintain the .NET framework on each end-user client, especially portable devices. Moreover, such .NET systems are tied to Microsoft's proprietary browsers and application servers.

Thin-client hardware is generally less expensive than thick-clients because they do not require as much local disk space, application memory or powerful processors. The total hardware requirement for a thin-client system (including both servers and clients) is usually much lower as well. One reason for this is that the hardware is better utilized. A thick-client CPU is idle most of the time. With thin-clients, memory can be shared. If several users need to access the LIMS application, it only needs to be loaded into the RAM once on a central application server. With thick-clients, each workstation must have its own copy of the LIMS in memory. Ultimately, the real value of thin-client computing emerges when determining the "total cost of ownership" (TCO), which includes not only the upfront price of the hardware and software, but also the much larger cost of installing, supporting and updating the system over time.

### **Selecting the right solution**

While the advantages of a thin-client strategy appear to be greater than that of a thick-client, there are many variables to determining the right solution for a particular organization. For example, consider a LIMS at a contract research organization (CRO) that would like to provide secure Internet access for data entry, analysis and reporting to external customers. It would be challenging to provide this with thick-client LIMS. Also, be aware that just because the application runs on a browser, it doesn't mean it is thin-client. Are your external customers running Internet Explorer with .NET so they can support access to your .NET LIMS? Do they allow .NET applications to be downloaded to their end-user clients?

Another consideration is to determine if the technology behind the LIMS is associated with a single vendor and if that raises concerns with your IT organization. For instance, does your IT organization want to limit itself to technology proprietary to Microsoft? This means that the LIMS most likely can only utilize Microsoft browsers and Microsoft back-end web and application servers. In this case, an existing Linux or Unix environment would not be able to run the .NET LIMS, thus complicating assessment of the actual cost for using such a solution since it may preclude use of existing technology.

It may also be important to consider whether remote handheld is critical to the lab. If so, see if the thick-client LIMS can run on a portable device and assess whether the necessary .NET support browser is available for PDAs and other handheld devices used in the lab. This can be a significant issue to organizations that conduct R&D in the field or perform activities on the plant floor, such as environmental monitoring.

### **Summary**

There are significant differences between thick-client, web-enabled, web-based and thin-client solutions. Thick-clients are desktop client applications that connect to a remote database. Web-enabled solutions are more oriented toward thick or rich clients, while web-based solutions offer some of the benefits of thin-client applications. Thin-clients, on the other hand, are true HTML applications that do not require additional end-user client downloads, applets or frameworks. Just be sure you know what you're getting by being cognizant of the differences, advantages and disadvantages of each type of solution in relation to your existing or planned infrastructure. Many organizations are choosing thin-client solutions and paving the way toward a simplified infrastructure that delivers lower cost of ownership, better usability and less complexity. Make sure the LIMS you select is the best fit for your organization.

1 <sup>1</sup>See "An Introduction to LIMS" by Helen Gillespie on [www.LIMSource.com](http://www.LIMSource.com) and "A Brief History of LIMS" by Dr. Gerst Gibbon,

published in Laboratory Automation and Information Management Issue 32, 1996, pages 1-5.

*Keith O'Leary is the Director, Product Marketing of LabVantage Solutions,*

**Copyright 2012 Scientific Computing, Advantage Business Media**