



PHL-LIMS Functional Implementation

By

Robert E. Bostrom

Kansas Division of Health &
Environmental Laboratories

2004 PHIN Stakeholders Conference

There are Fundamental Differences between PHL's and Other Laboratories

- Data is different
- Specimens/Samples are different
- Communication is different

Data

- Other Clinical or Environmental Laboratories analyze specimens to produce a result of analysis.
- PHL's produce results as well, but may collect additional epidemiological data which is irrelevant to producing the analytical result.

Specimens/Samples

- Clinical laboratories analyze clinical specimens
- Environmental laboratories analyze environmental samples.
- Food laboratories analyze food samples
- Animal laboratories analyze animal specimens
- PHL's analyze clinical, environmental, food and animal samples

Communication

- Other laboratories report to clients, usually in a standard format established by the laboratory
- PHL's report to the public; doctors; hospitals; law enforcement; local, state and federal agencies, etc. usually in a variety of reporting formats sometimes established by the laboratory, but more often by the report recipient

What is a PHL to do?

- Traditionally, PHL's have adapted in response to criteria presented dependent on logistics and software and hardware capabilities
- Implementation has been difficult, expensive, inconsistent and can require constant monitoring and maintenance

What is a PHL to do? (cont)

There are really only three solutions (or combinations of these solutions)

- COTS – modify an existing commercial application
- Outsource – hire someone to develop or modify an application
- In-House – develop or modify an application within the laboratory

Implementation



Currently Available Resources

- LIMS Requirements
- LIMS Design

What's Next?

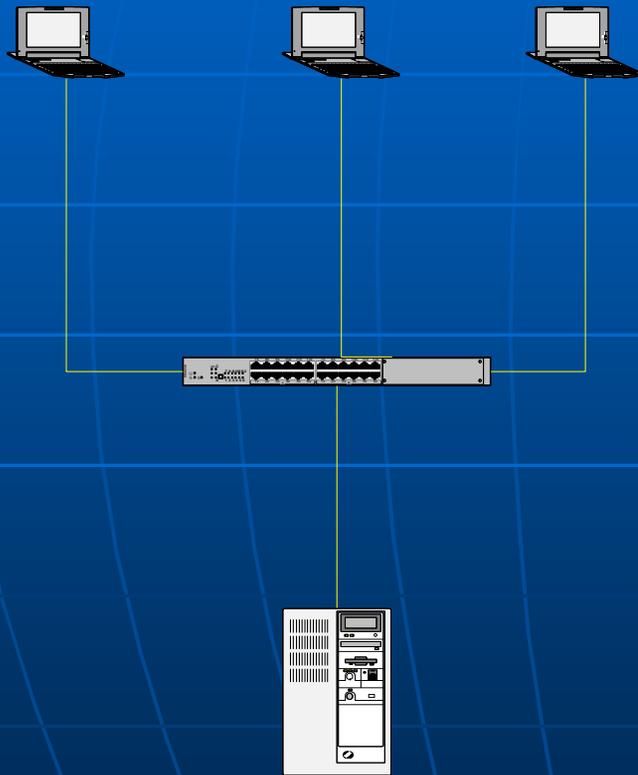
Regardless of whether this is a new implementation or a modification of an existing system, these are the system goals:

- The system and its framework must be secure
- The system must be scalable
- The system must support multiple databases
- The system should be applicable to large and small laboratories and relatively easy to maintain
- The system should support multiple OS architectures
- The system and its underlying language must be flexible
- The system must be capable of communicating in a variety of formats
- The system must be capable of using multiple front-end clients

Model Implementation

- Assume system is based on the Requirements Document and LIMS Design Document
- Assume New System
- Assume system is not a modified COTS
- Assume cost is not a major factor
- Assume there is an IT staff capable of providing support

Current Architectural Framework



- Most PHL-LIMS today are based on a 2-tier, client server architecture
- Clients access the database directly
- Client supplies the applications and business rules; Server supplies the database

Clients

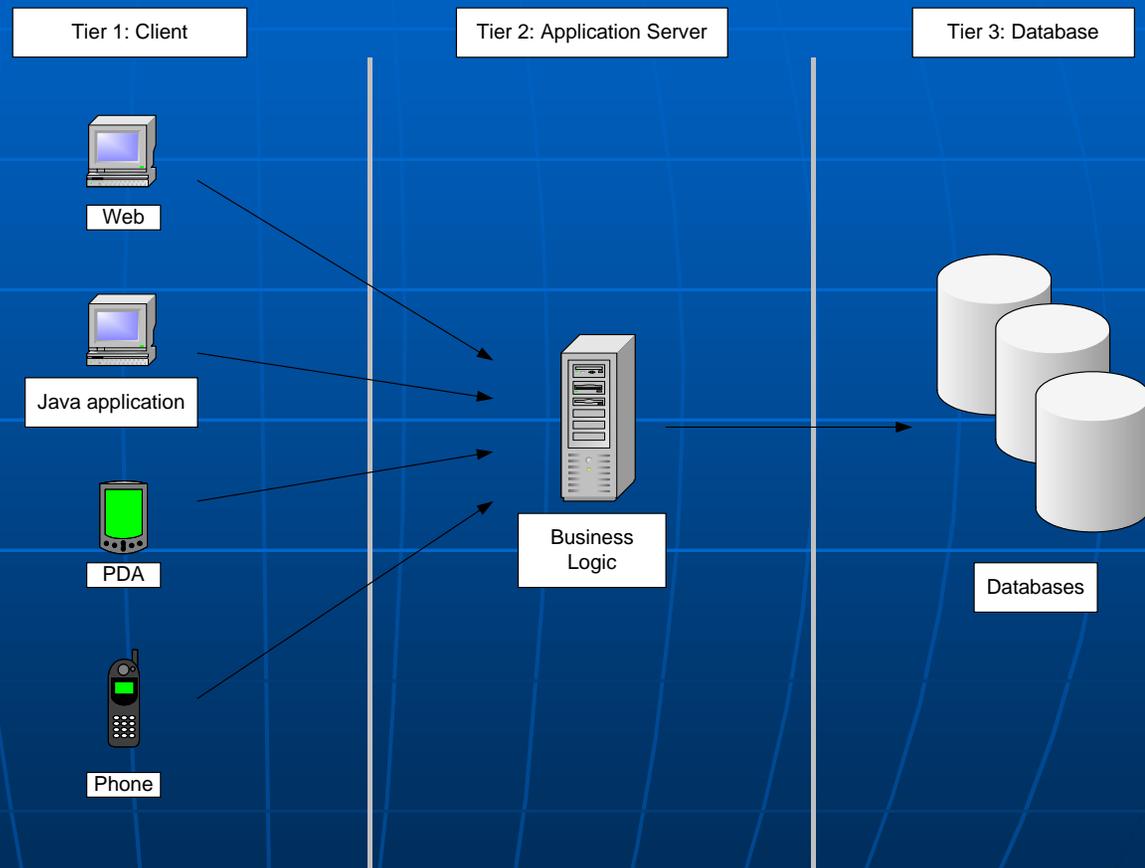
Limitations of the 2-Tier System

- Performance deteriorates as the number of users increase and the system is not scalable
- Restricted flexibility and choice of DBMS, since data language used in server is proprietary to each vendor
- Limited functionality in moving program functionality across servers
- Security is a problem
- Adaptability and scalability is limited

Architectural Framework (cont)

3-Tier with Application Server

- Clients "talk" to the Application Server directly
- Application Server supplies business rules and applications and "talks" to the database
- Database server supplies data



General Benefits of 3-Tier (N-Tier) Architecture

- Created to address the limitations of the 2-tier system
- Incorporates a middle tier between the user interface and the database which contains the business logic. It also allows the formation of an N-tier system where N-tier may incorporate additional business logic layers

General Benefits of 3-Tier (N-Tier) Architecture

- Decrease in the size and complexity of client programs
- Service and report components execute at the speed of the Application Server not the client
- Cache and control data-flow for better performance
- Less software maintenance

Encapsulation

- Only functions that are useful to clients are exposed
- Clients don't have to worry about the internal working of the LIMS
- As such, the client code is reduced in complexity
- The client code is the only code that is implemented multiple times (web, PDA, client access)

Reusability

- Business logic is written only once
- Reduces duplication of effort
- Prevents variation in business logic
- Enforces the consistent application of business logic
- Significantly different than traditional client server applications that store their business logic in the database since they have to re-implement their logic interpreter on every client

Security

- Each client must first pass its login information (credentials) to the application server before it can use any of its services
- Security is maintained in the application server
- The application server is responsible for allowing the user access to information
- Framework is secure because the system enforces security constraints in the application server, not the client
- The user cannot “touch” the application server
- Client can only access the database through the application server

Security (cont)

- System does not have to implement authentication, just authorization. Existing systems (i.e. Kerberos, LDAP or Active Directory) offer much more functionality
- Complex secure systems such as biometric or secure certificates can be implemented only once
- Users are not required to remember another password

Scalability

- The entire framework can run on any number of servers
- Small laboratories can start with one server that houses the client, application server and database
- Larger laboratories, based on throughput or geographic need, can deploy multiple application servers and/or databases
- Allows handling of data sources and interfaces distributed across several machines or even several networks which may be in different locations
- All configurations can be achieved without altering any application or client code

Language

- LIMS applications should be written in a standard, mature, fully-functional, multipurpose, object oriented language which can run under multiple computer operating systems on top of multiple databases (i.e. Java)
- Java incorporates advanced libraries which make it an ideal platform for writing large-scale applications
- Java can run on any OS that has JVM interpreter such as UNIX, Macintosh, Windows, etc.
- Java can access any database that has a JDBC or ODBC connector, which includes almost all popular databases (i.e. Oracle, SQL, Sybase, MySQL, etc.)

Clients

- A client is defined as any user, instrument or software that attempts to interface with the application server
- Clients may be individual users, PDA's, phones, the web, instrumentation, or external software not part of the framework
- Every client, regardless of type, goes through the same business logic

Instrument Gateway

- A client which provides the means of transferring data between LIMS and analytical instrumentation
- Clients are written which understand instrument commands rather than mouse/key commands
- Function varies by instrument
- Communication is bi-directional

Communications Gateway

- A client which provides the means of transferring data between LIMS and the outside world
- XML should be the preferred communications language
- The client will periodically look for information that needs to be transferred, creates an XML data package and sends the package to a translator
- The translator uses pre-programmed rules to rewrite the package to an acceptable format for the partner
- The package is then sent to be encrypted and delivered to the designated partner
- PHIN-MS is an excellent and preferred method of encrypting and delivering data
- The opposite scenario occurs when data is sent to the LIMS

Communications Gateway Using the World Wide Web

- Plays an important role for small to mid-size clients who do not have a full-blown translator
- These clients traditionally do not have information systems that can directly interface with state PHL-LIMS
- These clients can use the web to order tests, search for samples and pick up results
- Security is achieved using SSL (secure socket layer) for communication
- Authentication can be ensured using one-time passwords (i.e. SecurID) or personal secure certificates

Thanks!!

For additional information contact:

Robert E. Bostrom
KS Div. of Health & Env. Laboratories
Forbes Field, Building 740
Topeka, KS 66620-0001
Phone: (785) 296-1654
Email: bbostrom@kdhe.state.ks.us



Dari Shirazi
University of Iowa - Hygienic Lab
Oakdale Research Campus, OH-E5B
Iowa City, IA 52242
Phone: (319) 335-4500
Email: dshirazi@uhl.uiowa.edu