

"SOS - Software as a Service: The Evolution of Software & Tools becoming available over the Web for Lab Systems including Instrumentation and LIMS

Bill Pingpank, *VP Client Services*

Presentation Overview

- The laboratory software history
- SOS Definition
- SOS Current Users
- SOS Deployment Models
 - Dedicated
 - Semi-Shared
 - Shared-Separate Tables
 - Shared-Single Set of Tables
- Does it work for the laboratory environment?

Laboratory Software: The history

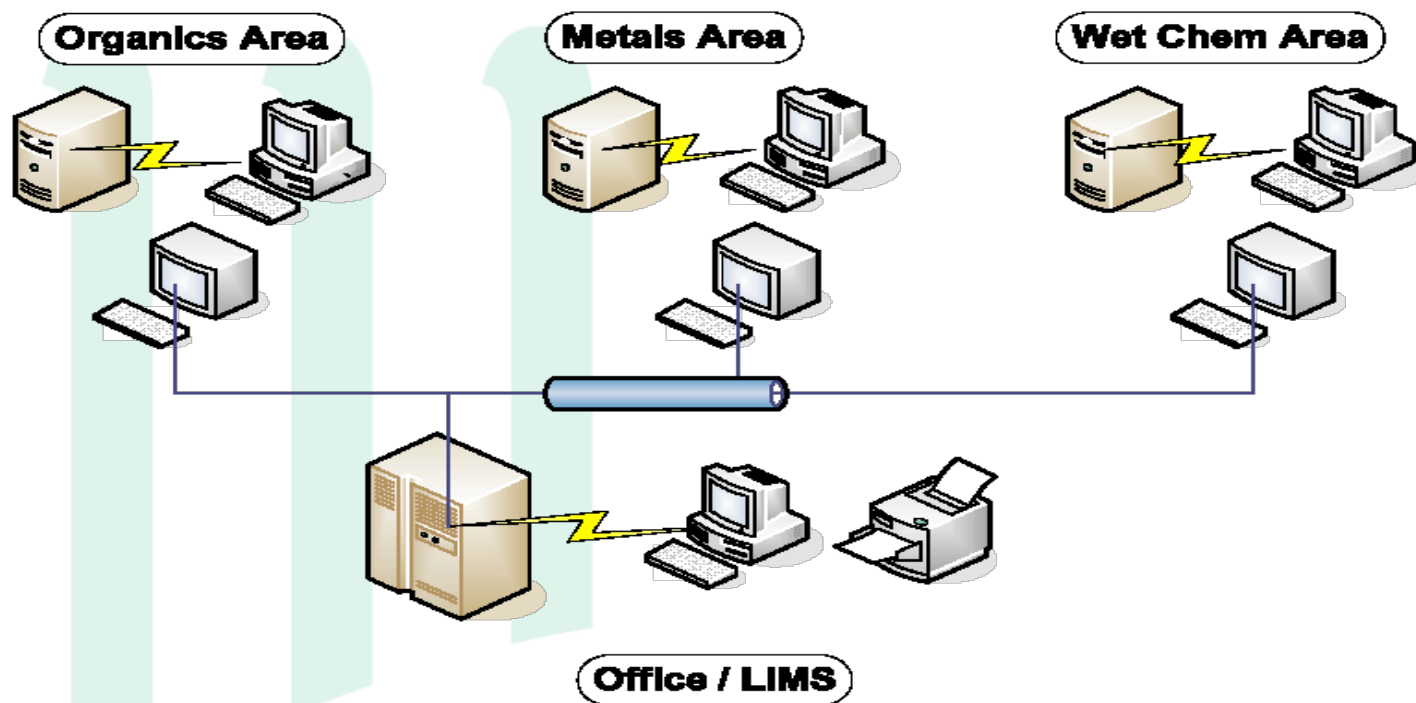
- Most Common forms of Lab Software
 - Instrumentation
 - Desktop Based (i.e. Excel)
 - Laboratory Information Mgmt Systems (LIMS)

Software Components

- Software can be broken down into 3 components
 - User Interface
 - No matter where your application is running you have to have this in front of you
 - Data Storage
 - Component rarely seen by end users
 - Good candidate for off site location since its rarely seen
 - Reporting Engine
 - Component rarely seen by end users
 - Also a good candidate for off site location

History of Data Management: Terminal

- Stand Alone Instrumentation/Terminal Entry

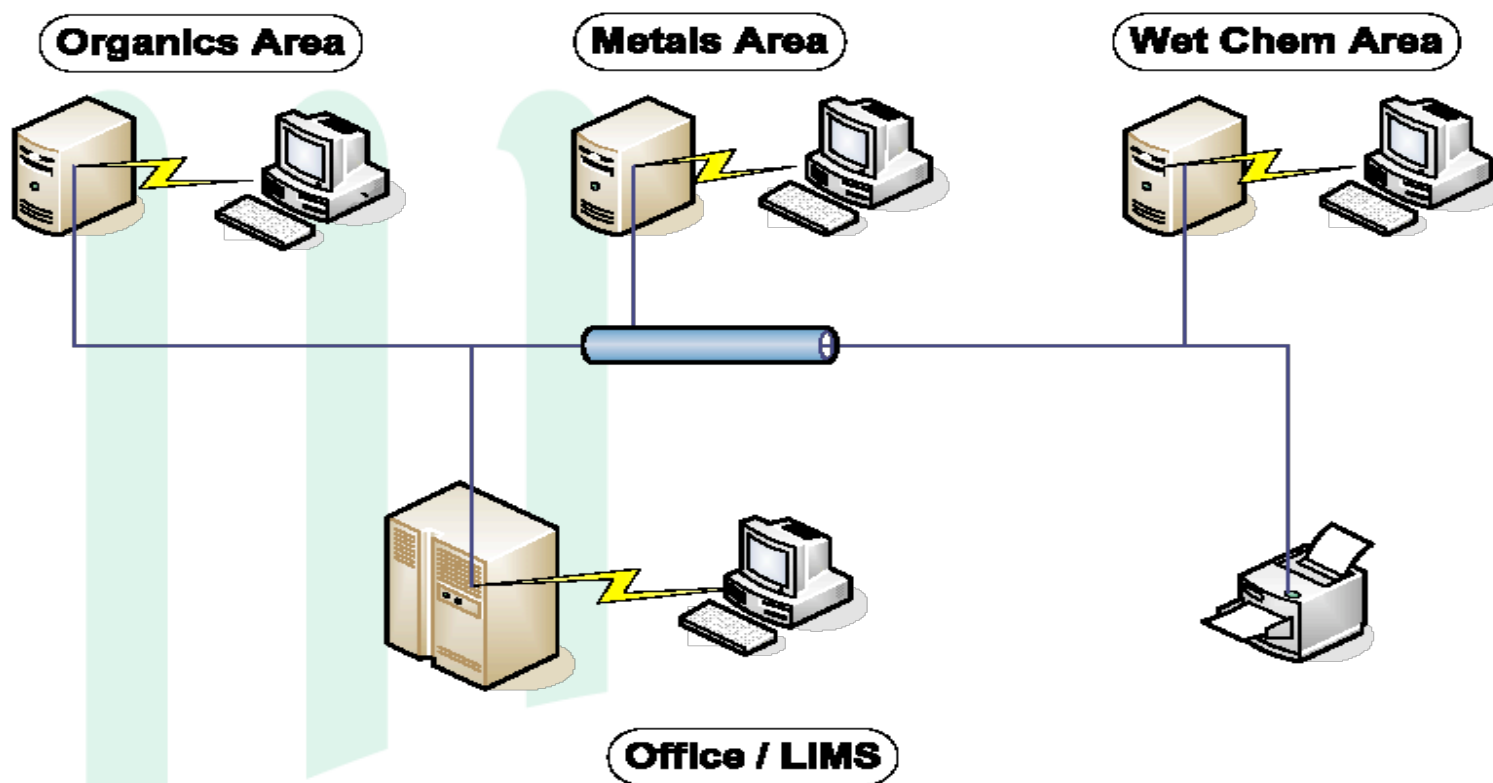


Pros and Cons of Stand Alone and Terminals

- Pros
 - Terminals were Inexpensive
 - At least in comparison to PC based computing
 - Allows for distributed Data Entry and Access
 - Data Centrally stored on Server
- Cons
 - Terminals were Dumb, unable to perform any functions or programs
 - Instruments were unable to communicate with terminals so you still needed a PC
 - This required data re-entry and increased potential error
 - Requires a server and connectivity within the operation

History of Data Management: Client/Server

- Client / Server Diagram

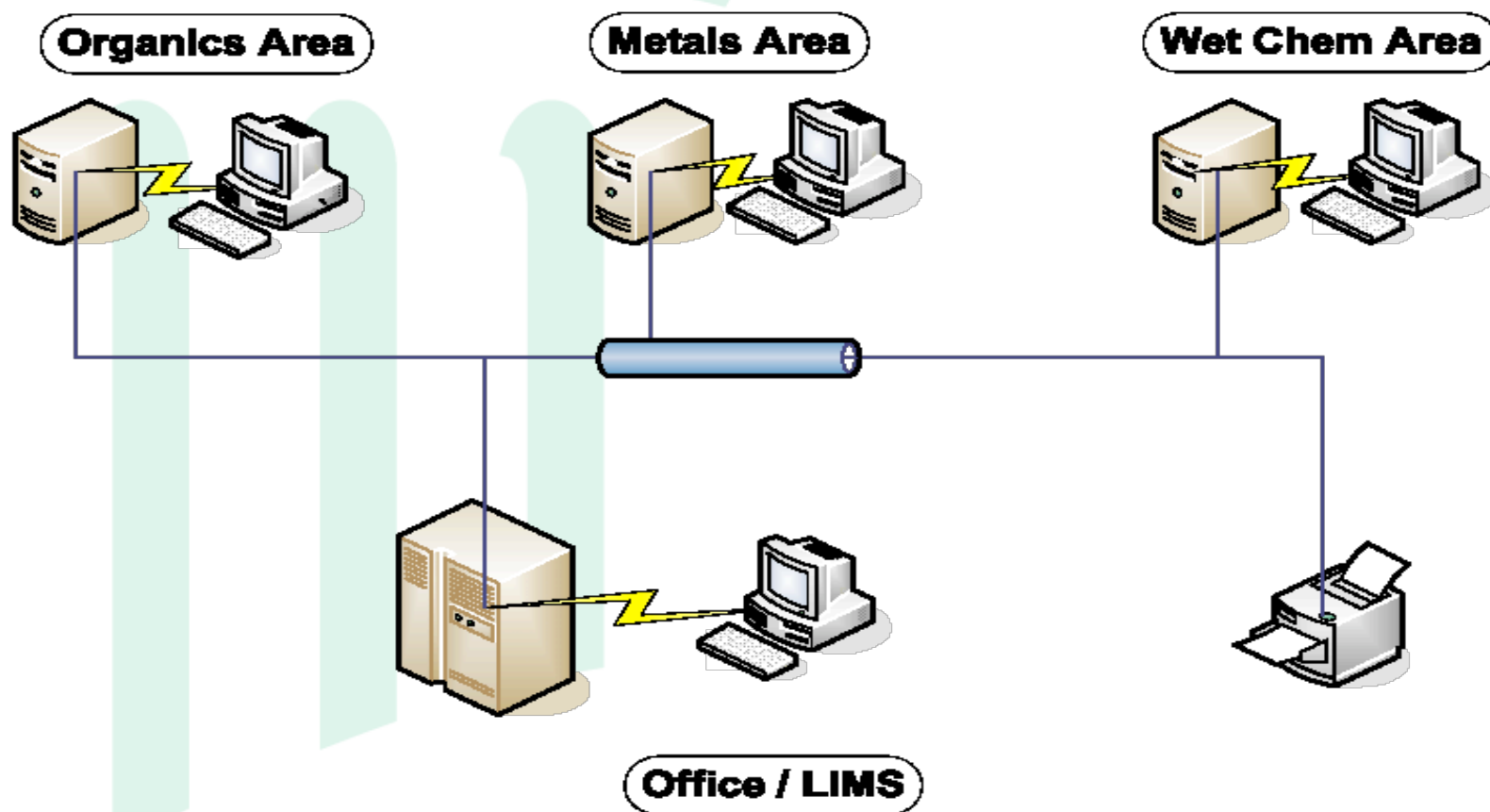


Pros and Cons of Client / Server

- Pros
 - PCs have become very inexpensive
 - Distributed processing power
 - Instrumentation and PCs could communicate over networks
 - Numerous Programs available for network or client specific install
- Cons
 - Have to install programs on the PC
 - Occasionally lead to software incompatibility
 - Updates require deployment to clients which can lead to incompatibilities
 - Security information often resides on the Client
 - Requires a server and connectivity within the operation

History of Data Mgmt: Thin Client/Server

- Thin Client /Server Diagram



Pros and Cons of Thin Client / Server

- Pros
 - PCs have become very inexpensive
 - Distributed processing power (may be limited)
 - Instrumentation and PCs could communicate over networks
 - Numerous Programs available for network or client specific install
 - Potentially no client installs/updates as it uses the browser for its interface.
 - No stored security credentials on the client.
- Cons
 - More server intensive processing as client processing may be limited
 - Still Requires a server and connectivity within the operation

Enter SOS

- SOS: Software as a Service is software solution that resides outside of your local network
 - Other terminology
 - Cloud Computing
 - Hosted Solution
- Very similar to thin client / server
 - Same browser based client
 - Server does not exist within domain

SOS Users

- More than likely, everyone currently uses SOS.
 - Individual Users
 - Hosted email
 - Hotmail, Yahoo, Gmail
 - Financial Management
 - On-Line Banking
 - Brokerage Accounts
 - Corporate Users
 - Salesforce.com

Initial Expectations of SOS

- Advantages

- Cost

- Possibly no hardware to purchase/install/network
 - No IT overhead to manage the servers

- Fast Deployment

- As soon as you have internet connectivity you can use the application

- Disadvantages

- Where is my data?

- Storage/backup/long term access

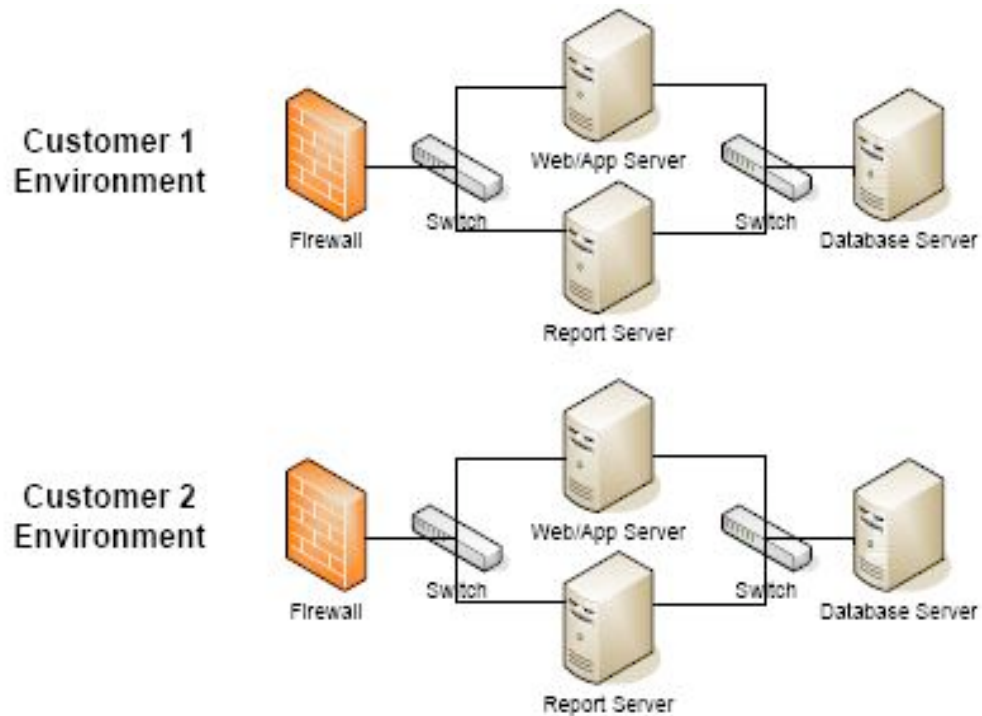
- Is it really as simple as going to a address somewhere on the web?

SOS Deployment Models

- Hosting Models
 - Dedicated
 - Semi-Shared
 - Shared-Separate Tables
 - Shared-Single Set of Tables

SOS: Dedicated Host

- A separate set of servers is setup for each customer. These are the same set of servers that the you would have installed internally. The administration is done by locating facility. Also called Co-Locating (CoLo)



Dedicated Host Pros and Cons

- **Pros**

- Customers avoid having to administer servers
- Customers can control the timing of upgrades
- Customers can control the timing of patches
- Easier to support application customization

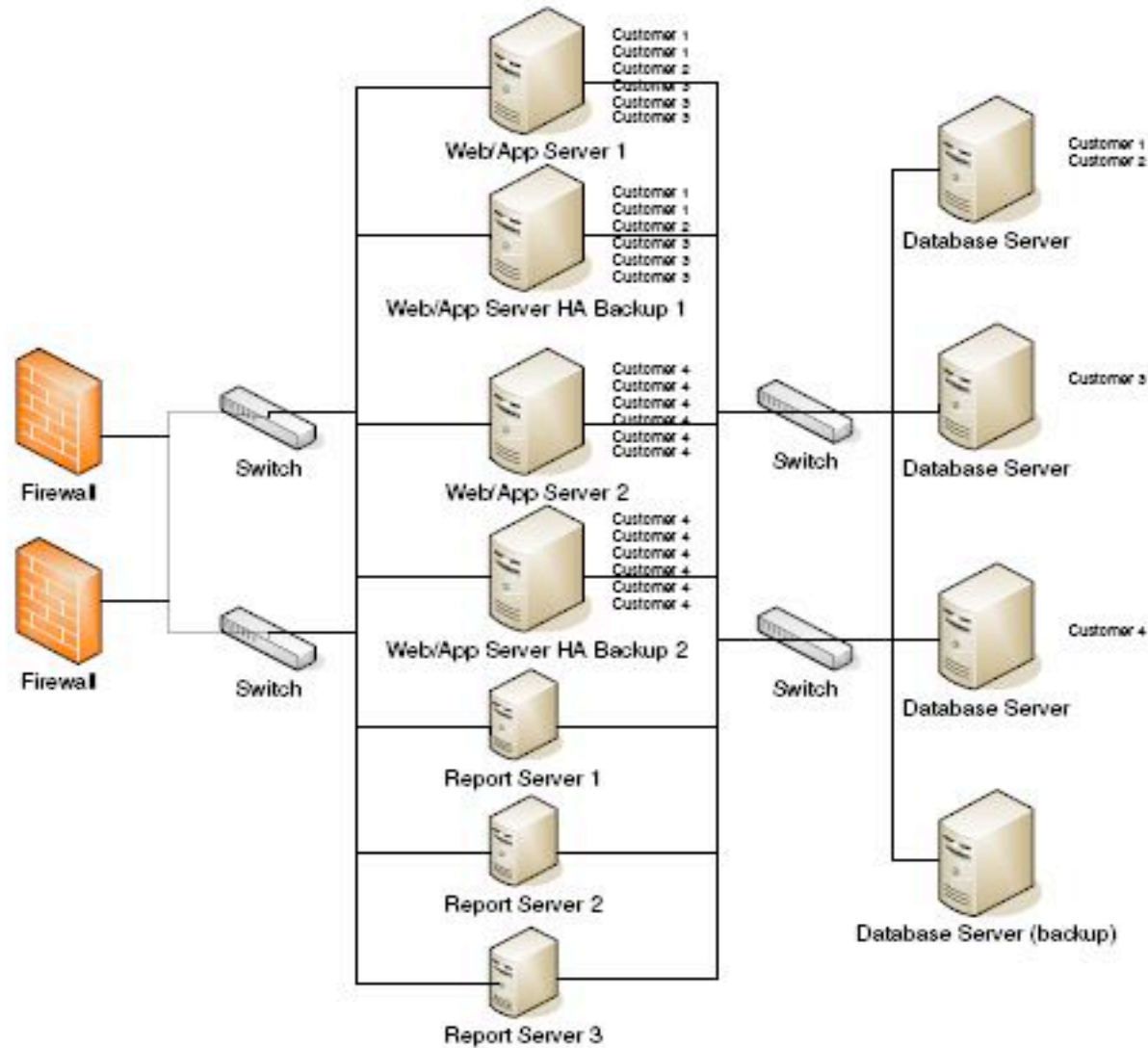
- **Cons**

- The Price/Cost is Very High
 - Same Servers + Data Center + Server Hardware Administration + Server
 - App Administration + Margin = Very High Price
 - Not an economical solution
- Scalability is dependant on hardware
- Potentially Subject to reliability problems

SOS: Semi-Shared

- Servers are shared by a few customers.
 - usually less than 6 per server
 - A Separate application instance or virtual servers for each customer.
 - Database servers may be shared based on customer loads

SOS: Semi-Shared Diagram



Semi-Shared Pros

- **Pros**

- Slightly more efficient than the Dedicated model leading to a decreased cost
- Efficiency may allow for additional scalability
- Customers can control the timing of upgrades and application patches

Semi-Shared Cons

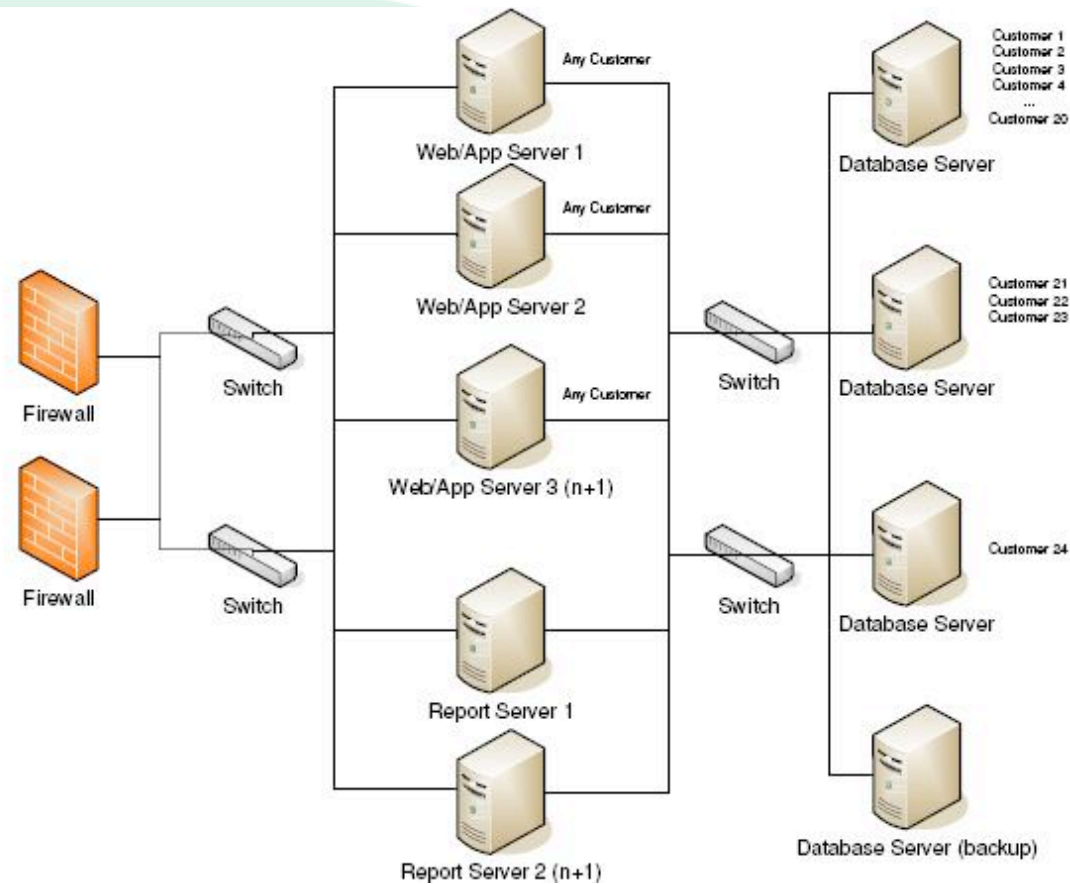
- **Cons**

- Most Expensive Shared options
 - New customers require new software installation
 - New Hardware
- More redundant servers are required for high availability
- Multiple software versions must be supported since each customer tends to install, upgrade and patch at different times
- Due to the diversity, there is less opportunity for automation increasing the risk for your site
- Much more administration manpower is required than with other shared models increasing the costs

SOS: Shared Separate Tables

- Customer environment is held in a separate set of database tables
- Application servers are shared across all customers on the same version of the application
- Database servers are shared based on customer loads
- 1 to 50+ customers per database server routinely
- Applications must be coded to support a multi-tenant model allowing a single application instance to concurrently connect to many separate data sources

SOS: Shared Separate Tables Diagram



Shared Separate Tables Pros

- Pros

- More efficient than the Semi-Shared model
- Efficiency allows for addition of High Availability and Scalability
- Allows for a significant amount of admin automation reducing costs
- Requires very few administrators
- Allows for support of just a few versions, usually 3 or less
- Customers can control the timing of their upgrades
- Avoids customer security concerns over commingling data in the same set of tables

Shared Separate Tables Cons

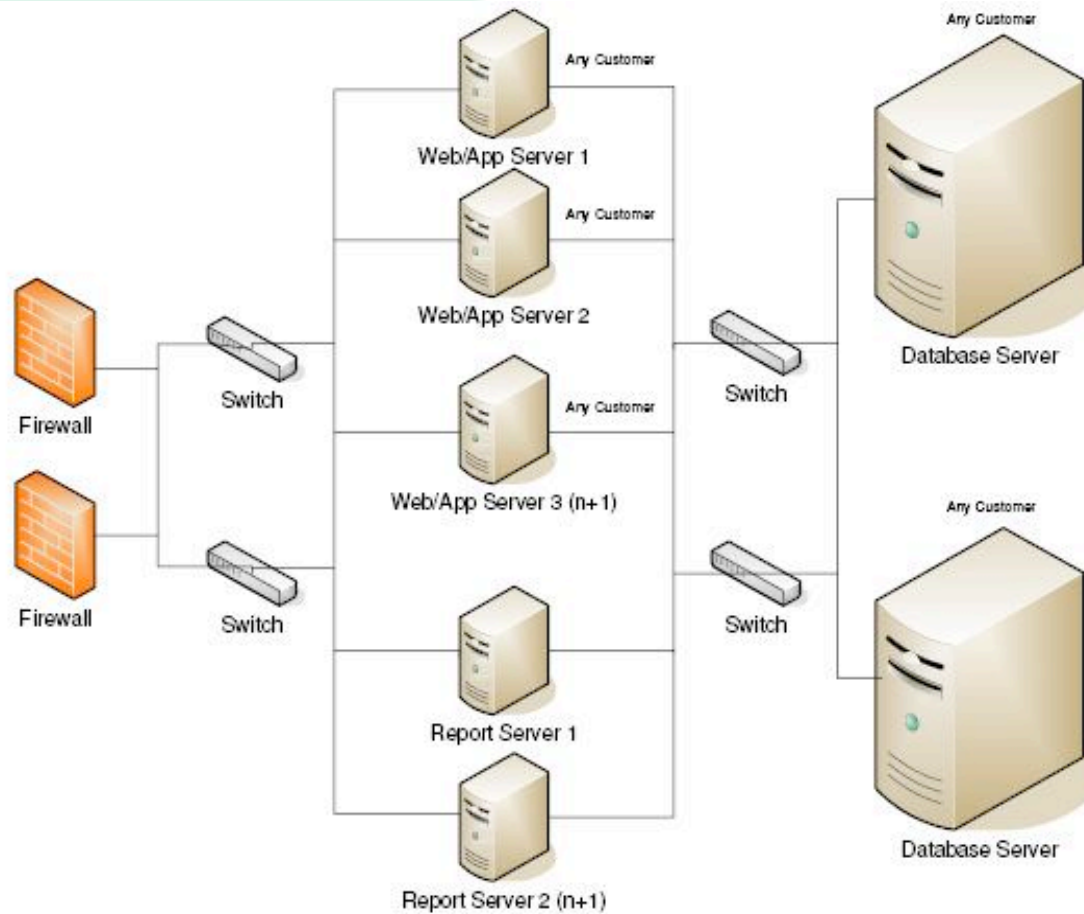
- **Cons**

- Multiple versions are supported making support more difficult
- Not as efficient as having multiple customers in the same set of tables
- For many applications, customization can be difficult or impossible to support.

SOS: Shared Single Table Set

- Application servers are shared across all customers
- Single Database shared by all customers
- Applications must be coded to support a multi-tenant model supporting multiple customers in the same set of tables

SOS: Shared Single Table Set



Shared Single Table Set Pros

• Pros

- Most efficient model for the host leading to the least expensive option for the Laboratory
- Efficiency allows for true scalability at the lowest cost
- Allows for a significant amount of admin automation reducing costs and improving QA
- Requires fewest administrators
- Allows for support of just one version keeping code base small reducing error potential

Shared Single Table Set Cons

• Cons

- Customers may have security concerns over data being held in the same set of tables with other customers
- Requires ever larger database servers and processing power, at a significantly higher cost
- Upgrading customers all at one time carries higher risk
- Development and QA processes have to be tightly aligned with this model
- Customers have no control over the timing of upgrades
- Customization is normally impossible to support.
- Backing up data uniquely for customers is more cumbersome in more complex applications – laboratory data is a good example

SOS: Does it Solve Lab Issues?

- It Does:

- Reduce the price to access advanced applications opening the door to smaller laboratories to use very sophisticated software.
 - Facilities will be able to meet stricter regulatory requirements without having to mortgage the lab.

- It Also Does:

- Reduce the access to your data because its off site.
- Inherent complexity of laboratory data makes it difficult to get useful Backups of your data.
- Application is exposed to significantly more variables than if it were installed in a pure local environment.
 - If lab internet connection goes down so does access to the entire application
 - Web hacking and vulnerabilities exist

Questions and Answers

- Q&A