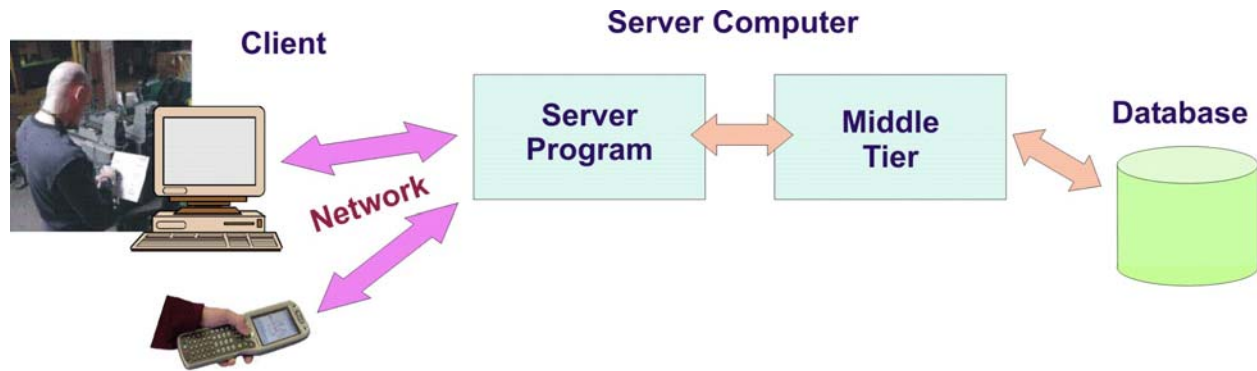


## Thin versus Thick Client Architectures for Tracking Industrial Operations



### Introduction

This monograph was written to provide some thoughts about whether an organization should go with a "thick client" or a "thin client" architecture when implementing a software application to track jobs, materials, labor and equipment in real-time in an industrial setting.

A thin-client architecture is normally used to refer to the situation where the user of the application uses a web-browser on a Personal Computer (PC) or Mobile Computer (MC) to connect to the material tracking application over the Internet or a local area network or a wide area network. It can also refer to a "Citrix" or "Terminal Server" situation where the user's personal computer (PC) simply runs screen display and keyboard utilities and the main application is run in a central server computer. A thick-client architecture usually refers to a client-server architecture in which a program running on a PC or MC handles all the interaction with a user and then relays the data to a database server.

Applications are said to run in terminal mode when the client acts as a simple terminal to the server. Terminal mode applications require constant communication with the server. This is in contrast to store-and-forward mode clients which only require intermittent communications to transfer their data. All thin-client applications use terminal mode and require continuous communications with the server.

The discussion of the trade-offs here is focused on the data collection, barcode scanning and barcode label printing aspects of the systems. BellHawk Systems has implemented data collection systems using both types of thin and thick system architectures. We have also implemented systems that use thick clients for data collection and web-based access for all the users who simply wish to look at the tracking data. For most of our mid-market industrial clients, we use thick clients for the data collection PCs and store-and-forward technology for the data collection mobile computers as we find these the most effective for the reasons described below.

## Architectures

We first need to recognize that all these thick and thin architectures are fundamentally the same. They have a client program running in a PC or MC that sends and receives packets of information over a network to a server program that communicates with a middle-tier software program that stores or retrieves information from a database. The middle-tier is that part of the back-end application where most of the processing of information occurs prior to storage of information in the database. It also assists in efficient retrieval of information from the database.

The following table gives a comparison between four types of application. All these examples use Microsoft technology, which is the most pervasive in the mid-sized manufacturing companies that we serve.

Architecture	Client Program	Server Program	Middle Tier
Thin Client	Internet Explorer running on PC or MC	Microsoft IIS Web Server	Server Application written in ASP.net
Thin Client	Terminal Server Client running on PC	Microsoft Terminal Server	Client Program written in VB.Net
Thick Client	Access/VBA Program running on a PC	Microsoft Network Services	SQL-Server Stored Procedures
Thick Client	VB program running on MC	BellHawk Store and Forward Wireless Server	SQL-Server Stored Procedures

## Advantages and Disadvantages

### ***Web Based (Thin Client)***

Advantages:

1. Quickest and easiest to develop with a team of experienced programmers.
2. Quickest and easiest to deploy for IT departments as there is no special software to be installed on any of the PCs or MCs.

Disadvantages:

1. No end-point data validation in a web-browser. A web-browser will take in any data entered into a screen form with no validation as to its format or contents. In capturing data from the factory floor or warehouse it is essential to validate each data item scanned or entered because of the low literacy level of the users. In a browser-based architecture all this validation has to be done in the mid-layer after sending a filled out web form to the server. This leads to some nasty choices:

- a. Validate after capturing each data item. This leads to long delays and slows down workers. Just go onto the Internet and time how long it takes for a website to respond. Now put that time between each scan of a barcode in putting away materials in a warehouse or scanning materials into a job and you will get some idea of the issues. I have personally observed wireless connected PCs take as long as 20 seconds between scans to retrieve web pages from a server.
  - b. Wait until the user has filled out a form with all the data before sending it to the web server. If a user makes a mistake then they will have to enter the data again. There is no immediate corrective feedback.
  - c. Download a Java applet to do the data validation. Because the Java applet needs access to the database, we have essentially created a client–server application with the disadvantage of a long–download time of the Java applet from the server.
2. Web browsers cannot use data–directed barcode scanning or keyboard data entry sequences. With a web–based approach, the data entry form must be predetermined at the start of data entry. It is not possible to do data directed scan sequences without downloading a Java applet, which turns the web client into a slow response client–server application. A critical element in making a material tracking system easy to use is to minimize the number of scans or data entry items. This requires data directed scan sequences, which require access to the system database (or a copy of the database).
3. Web browsers have to present user with a form with multiple data entry boxes. In our experience, typical factory and warehouse production operators frequently enter data into the wrong boxes. We have found that the only way to minimize mistakes is to present users with only one data entry box at a time and then to validate that they entered the correct data before proceeding with the next data entry item.
4. The web browser needs continual communications between the users PC or MC and the Webserver. If the communication is over a local area network then this is usually not a problem. If the communication is over a wireless link then this can be highly problematic in a factory or warehouse. WiFi radio waves are absorbed by metal racking, equipment, conductive materials and people. They are interfered with by electrical splash noise from equipment and welding operations. Even with many access points and antennas it is very difficult in many plants to maintain continuous communications. The result is lost data and the inability to perform needed operations in a timely manner.
5. If the server goes down or the communications network is broken then data collection stops.
6. Using a web browser approach, it is difficult to customize any of the client stations to the specific need of the manufacturing situation.
7. A web browser cannot interface to any factory floor equipment such as weighing scales, length counters, and process control equipment to automatically capture data unless a Java applet is downloaded into the client PC, through the web browser, to turn the browser into a slowly responding client–server application.
8. Difficult (if not impossible) to interface to other systems (in a services architecture) if the Webserver is not on the local area network as the other systems. Web servers run off–site,

such as by Application Services Providers (ASPs) have proven very difficult to make effectively communicate with ERP and accounting systems (if not run by the same ASP).

9. A high degree of programming skills necessary to customize the application to needs of specific organization due to the webserver programming skills needed.
10. Web browsers cannot interface to barcode printers attached to client PCs when the barcode printers use native printing mode for speed and special handling of labels, ribbons and tags.

### ***Terminal Server Based (Thin Client)***

Advantages:

1. Easy for IT Department to deploy; no need to install software on each PC.
2. Client programs can rapidly validate data as it is entered.
3. Clients can be programmed in Access/VBA so they are easy to customize.
4. Can use data directed scan sequences and can take in one item at a time with validation after each scan or data input.

Disadvantages:

1. Need to develop client–server application; but clients run on central server not on PC.
2. If the server goes down or the communications network is broken then data collection stops throughout whole plant.
3. Benefits of ease of deployment lost if any of the client stations are customized to the specific need of the manufacturing situation.
4. Cannot interface to any factory floor equipment such as weighing scales, length counters, and process control equipment to automatically capture data.
5. Difficult to interface to barcode printers that are not on same local area network as server.
6. Terminal server computers are very expensive as they have to enough processing power to simultaneously execute a copy of the client application for each user of the system. In our experience, the cost of a multi–processor server is several times greater than the cost of purchasing separate PC and a suitable database server.
7. Terminal server computers are also difficult to maintain. We had one client who accidentally shut down a manufacturing plant for 3 days while attempting to perform an upgrade to the Terminal Server software.

### ***Client Server Based (Thick Client)***

Advantages:

1. Robust technology that has proven to run 24x7 in many manufacturing plants.
2. Inexpensive to deploy; uses inexpensive PCs and database servers.

3. Immediate checking of captured data by client program running on PC. Can use data directed scan sequences, for ease of data entry, with validation and immediate feedback to the user after each data entry step.
4. One data entry item at a time with extensive validation before proceeding to next step.
5. If PC Clients are written in Access/VBA, they can be readily customized by manufacturing engineers for different work centers. They can also be easily interfaced to equipment such as weighing scales and process control equipment to automatically capture data.
6. If server side is written as SQL stored procedures, then operation can easily be modified by a manufacturing engineer or database administrator.
7. Can easily interface with barcode printers and RFID portals using native-mode printer drivers.

Disadvantages:

1. More expensive to develop than web based application.
2. More work for IT department to deploy and maintain software on multiple PCs.
3. Needs continual communication with server; works well for PCs over local area network but not for wireless MCs.

***Store and Forward Systems (Thick Client)***

These are usually used on wireless MCs but increasingly used on PCs for high reliability operation.

Advantages:

1. Does not need continual communications with the server for operation. Can work where there is intermittent communications and can work in the presence of high levels of interference. Can be used for route delivery, yard, or off-site work where there is no communication with the server for prolonged periods.
2. Immediate checking of entered data against local database; results in rapid operations.
3. Can keep functioning even if server is down.
4. Robust technology.
5. Less expensive to deploy than terminal server wireless systems; needs many less access points and antennas.
6. Data directed scan sequences with one data item input at a time followed by immediate validation of entered data against local database in MC.
7. If MC Clients are written in Visual Basic, then data collection sequences can be customized by manufacturing engineers.

8. MCs can be interfaced to equipment such as long-range scanners and weighing scales for use on fork-lift trucks.
9. If server side is written as SQL stored procedures, then operation can easily be modified by a manufacturing engineer or database administrator.

Disadvantages:

1. Much more expensive to develop.
2. More work for IT Department; IT Department needs to deploy software to each MC or PC.

## **Conclusion**

It is apparent that the choice of which architecture to use for PC deployment comes down to an established trade-off. The IT department would like everyone to do all their computing through a central computer as this makes their life much easier (I know, I used to run an IT department). Most users on the other hand, prefer to use software on their own PC for its responsiveness and control. I think that some IT departments would like everyone to do all their computing through a central Webserver (preferably run by an ASP) or failing that, run their applications on a main-frame (preferably painted blue) in a central "Glasshouse" only accessible to IT personnel.

The reality is that different organizations have different needs. In the mid-sized industrial organizations (that are most of our clients), I find that the use of a client-server architecture for PCs is the most beneficial. This is because it is relatively inexpensive to deploy, is very robust, and is easy to customize to the needs of the specific organization. Also for 20 to 30 or so PCs typically deployed in a plant for data collection purposes, the IT maintenance load is fairly minimal.

For smaller organizations, with little or no IT or manufacturing engineering support, then the web-based approach (preferably run off-site by an Application Services Provider) would appear to be the best solution. Our experience in offering such a service, however, led us to understand that the benefits of customization and system responsiveness often outweigh the benefits of minimal support requirements, even for small companies.

For large organizations, where the IT Department may have to support many hundreds or thousands of PCs, then a web-based approach may be beneficial (especially for the IT department) providing all the PCs can reliably communicate with the central Webserver. In this case, the expense of tailoring and maintaining the Webserver application to the specific needs of different parts of the overall corporation can be justified. It also means that IT can control what runs on each PC in its many plants. This does, however, bring with it the disadvantage of not being able to interface equipment to each PC station (without heroic efforts in Java).

When it comes to the deployment of wireless mobile computers in factories, warehouses, and other industrial environments with intermittent wireless communications, then the store-and-forward technology is proving itself to be much superior to terminal mode operation (especially in comparison to Webserver based applications).

On balance, for most of our clients, thick client applications (with client server for the PCs and store-and-forward for the MCs) seem to be the way to go, with the currently available

technology. But, from time-to-time, we do implement specialized thin-client applications (using web-based technology) for those clients who need to provide access to the operational tracking data to a large number of geographically distributed people.

### **Author**



Dr Peter Green is President and Chief Systems Architect for BellHawk Systems Corporation. He is an expert in the use of software to track, control, plan and schedule industrial operations. Dr Green holds a BSEE and a Ph.D. in Computer Science from Leeds University in England. Prior to founding BellHawk Systems, Dr. Green was a Professor of Computer Engineering at WPI and a research staff member at MIT. Dr Green has over 30 years experience leading teams in the development of systems for Government, industrial, and commercial clients including many mission-critical web-based, client-server, and distributed peer-to-peer systems.