From the Cloud to the Edge for Barcode & RFID Data Capture

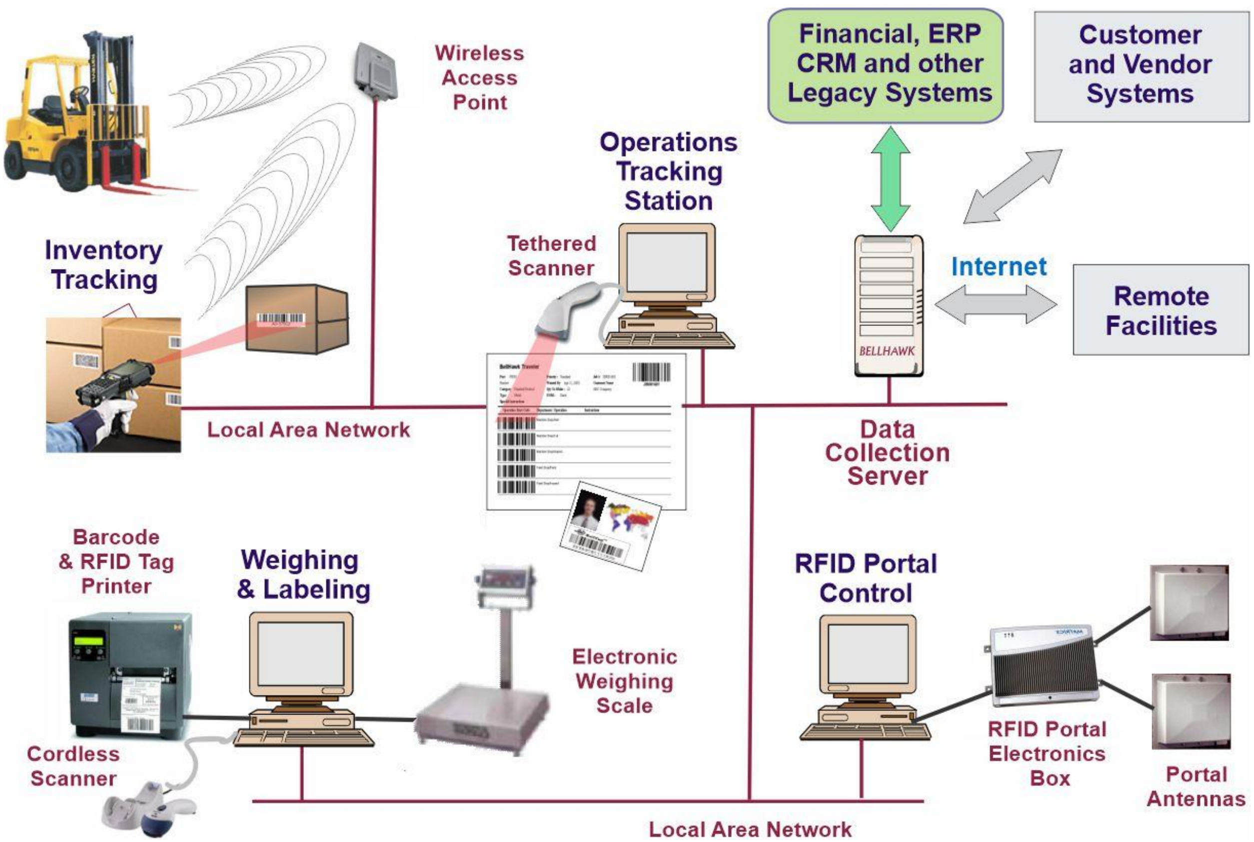
A White Paper by Dr. Peter Green

Introduction

In this White paper, we trace the evolution of the BellHawk Barcode and RFID job and materials tracking system from its early days using a client-server architecture, to running at a remote data center “in the Cloud”, to the newest evolution, using an Industrial Internet of Things (IIOT) Edge-Computing architecture.

We will look at each stage of the evolution of BellHawk, the forces that drove the change, and the advantages and disadvantages of each architecture, and how we can use a hybrid of Cloud and Edge computing together.

Early Days



BellHawk started out in the 2002/2003 time frame as a barcode tracking system, with user interface programs, written in Access and Visual Basic, running on Windows PC’s, with barcode scanners connected in parallel with the keyboard cable. Initially BellHawk used an Access database, but subsequently transitioned to a shared SQL database running on a Windows Server computer, accessed over the plant LAN.

This system started out tracking jobs through a sequence of operations, with materials tracking and warehouse management being added by Version 3, along with an interface to the BarTender software for printing out barcode labels on a wide variety of barcode label printers.

Version 3 also saw the addition of wireless mobile computers, each with their own thick client program, written in Visual Basic. Each mobile computer had its own local database, in which data could be stored until it could be relayed over the wireless LAN to the server.

The local database in each mobile computer also contained a simplified mirror of the BellHawk database, which was updated whenever the mobile computer could communicate with the server, and was used to provide point-of-action warnings whenever the user was about to make a data entry or operational mistake. These warnings could be provided without needing the user to be in wireless communications at time of scanning, which was essential due to the poor quality of wireless network communications technology available at the time.

Version 4 in around 2006 added a weighing scale interface to the Access program in the PC client computers. Version 4 also saw the first attempt to develop an external interface between external systems and the BellHawk database. This consisted of a set of Transact SQL procedures which simplified reading and writing the BellHawk database.

Finally, fixed station RFID portal scanning was added in V5 in 2008, using a custom program written in Visual Basic, running in a PC.

This system worked very well, was well-liked by its users, and still has some die-hard users who have been running this software, in one case, for over 20 years.

What many clients liked was that their IT staff could easily modify the Access/Visual Basic code, themselves, especially to create custom versions of standard reports.

So, why then, did my team develop a Cloud-based version of BellHawk, starting in 2008/2009?

Well, several things happened:

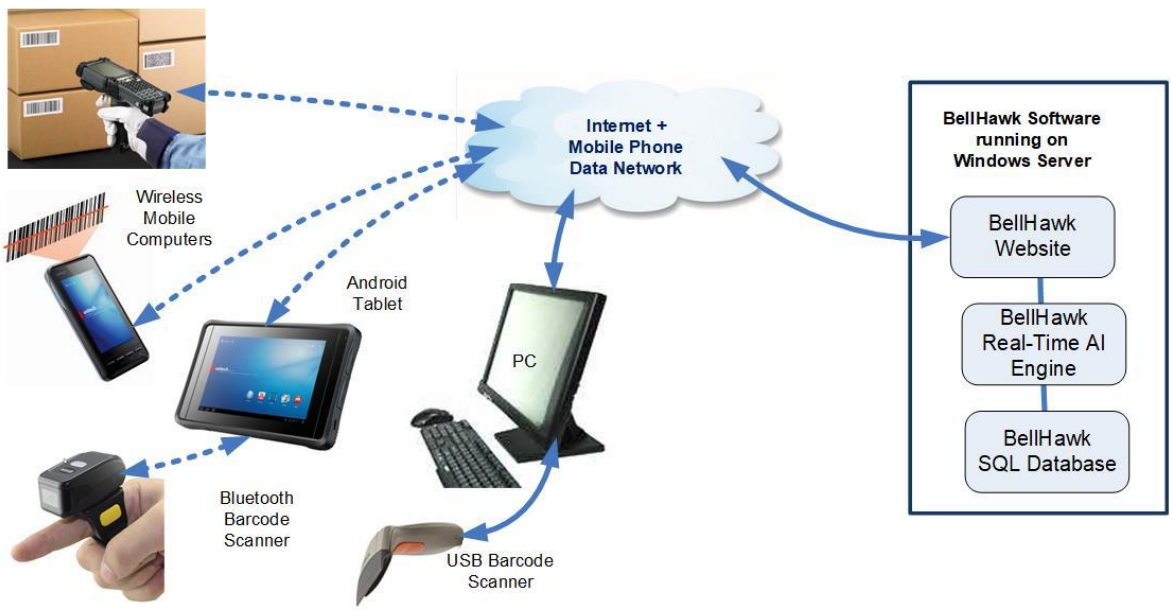
1. Microsoft decided that the client-server architecture was dead and that the future lay in the Cloud. As a result, they stopped supporting more and more of the features in Access on which BellHawk depended, with each new release of Windows. This required our clients to transition to the Cloud or continue using BellHawk on old computers, which posed a security risk.

2. The size of the IT staff in each manufacturing plant or distribution center dropped dramatically, typically from five people to one or none, due to cost cutbacks in the 2008 financial crisis (and has never recovered). Deploying and maintaining a BellHawk system required updating the software in each PC and mobile computer, which required a lot of IT staff time. Maintaining the local databases in each mobile computer so they were in sync with the main BellHawk database also proved to need a lot of IT support. This was because users would leave the mobile computers out of wireless communications range for a prolonged period of time, resulting in loss of database synchronization. The databases then had to be manually resynchronized if many missed overlapping changes had taken place in the BellHawk database and the local mobile computer databases.

3. Organizational consolidation. In the 2008/2009 downturn, many independent manufacturing plants and distribution warehouses were acquired by larger companies and formed into groups with a central IT support organization. These IT organizations wanted BellHawk to run in their own data center or on a commercial Cloud server. Also, they wanted to consolidate the information from all the plants and warehouses into their central ERP system. These requirements were not compatible with the use of a client-server data collection architecture.

4. Technology had improved to the point where Cloud computing was becoming readily available at an affordable price. Also, high reliability Internet was becoming widely available, including over wireless networks within plants and warehouses, and outdoors using a mobile-phone data network.

Transition to the Cloud



BellHawk V6 was the first Cloud-based version of BellHawk. It consisted of a specialized website, written in ASP.net connected to a SQL Server database. In the web-based versions of BellHawk, data capture is done using PCs and tablets with external barcode scanners and mobile computers with embedded barcode scanners. These devices all use a standard web-browser to communicate with the BellHawk website over the Internet, via the plant local area network (LAN), or wireless network in each manufacturing plant or warehouse.

This change to a web-browser interface eliminated the need to install or maintain software or databases on any data collection device. It also enables data collection to take place from anywhere there is an internet connection, including over the mobile-phone data network, using smart-phones equipped with external Bluetooth scanners.

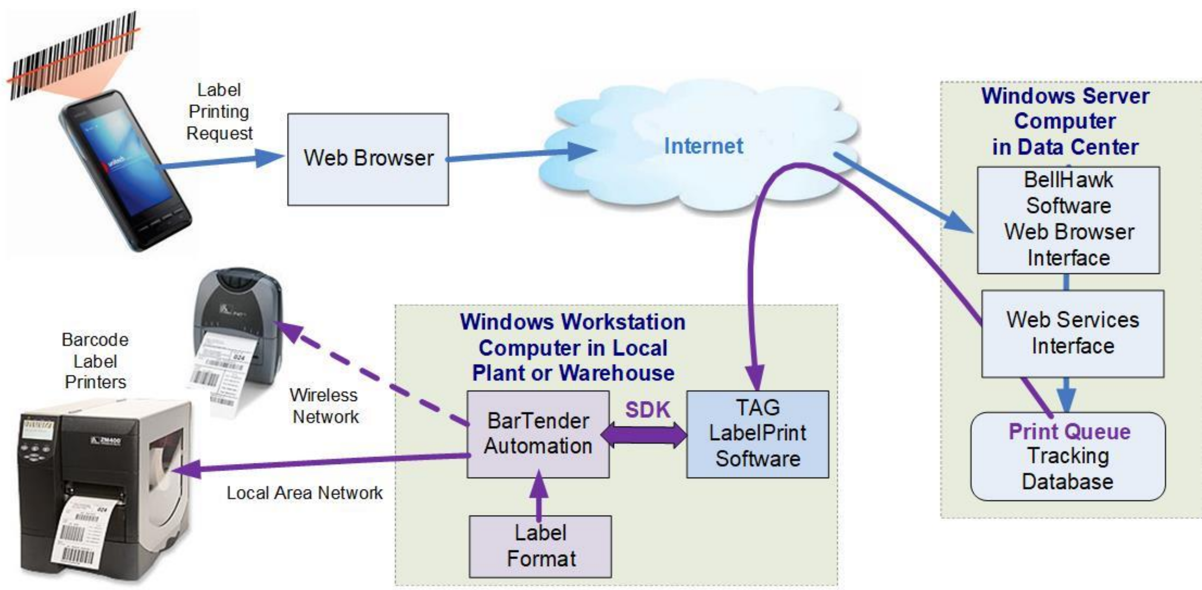
It also turned out to be the correct decision when a couple of years later Microsoft stopped supporting the Windows Mobile operating system on which our thick-client mobile computer software was based.

This change enabled BellHawk to now be installed in an organizations data center and supported by a centralized IT group, with no requirement for on-site IT support. It also enabled KnarrTek to offer the BellHawk Online service whereby smaller organizations could run BellHawk, over the Internet, on Windows Servers managed by KnarrTek running at a secure data center. As a result, clients were able to reap all the benefits of Cloud computing.

But, as in all things, there are trade-offs.

With the old client server system, the software doing the barcode label printing, in each PC, was able to rapidly read the BellHawk database, over the plant or warehouse LAN, to get the data needed to put on each barcode label that it sent to a printer. But, in a cloud-based system, the BellHawk database could be thousands of miles away in some data center. Also, the quantity of instructions that need to be sent to a barcode label printer, to get it to correctly print images, can be multiple megabytes of data. While this is no problem to send over a high-speed plant LAN, it can take a substantial amount of time to send directly to the printer over the Internet, especially over a slow Internet connection.

To solve this problem, we introduced the concept of a barcode label print server, running in each plant or warehouse, communicating with the BellHawk website over the Internet, by means of its web-services interface.



When a user is entering new material into BellHawk, they can select a button on the data collection device's screen and have a print request sent to BellHawk along with the entered data. After the data is stored in the BellHawk database, expert systems rules, stored in the BellHawk database, are then used to retrieve the label format, the printer-name, the data needed to put on the label, and the count of labels to be printed, and to place these in the label print queue in the BellHawk database.

The BellHawk TAG label print software, running in a Windows Workstation in the local plant or warehouse, then long-polls the BellHawk web-services interface over the Internet to retrieve entries from the print queue, as soon as they are available. As the amount of data to be placed in fields on the label is small, this can be quickly transferred to the workstation over the Internet.

The TAG software then uses the SDK interface to the BarTender Automation software to retrieve the label format and populate it with the required data before it is transformed by BarTender into the string of instructions to be sent to the appropriate printer. The resultant data stream can be quite large but can be transferred rapidly to the printer over the plant local area network or over the wireless network, if a wearable printer is in use.

This solves the problem of quickly printing barcode labels in each plant or warehouse even though the BellHawk server and database may be thousands of miles away.

With this configuration, users can use a monitor, mouse and keyboard attached to the workstation to create new label formats and modify existing label formats, which are stored locally on the workstation. What they cannot do is to modify the code that generates the data to be placed on each label, as they used to do, by modifying the Access/Visual Basic code in each PC that needed to generate a specific label format.

To solve this problem, we introduced the use of expert systems rules to select the data to put on each different label. These expert systems rules can be imported in the form of Excel spreadsheets by the BellHawk systems administrator through the web-browser interface, thus enabling an organization to set the label printing rules and formats themselves, without the need for any code changes.

Once the rules are setup, all the user has to do is select a button on their devices screen, to have appropriate labels printed out automatically, based on the stored expert systems rules. This eliminates much time wasted in using a separate label printing program and also eliminates errors due to the user “fat-fingering” in the wrong data into a label, prior to its printing.

The user is not aware that the print request is being relayed through a server thousands of miles away, unless the Internet or the BellHawk server is running too slow, which brings us to the next issue, which is server response time.

When a user scans a barcode or requests a label to be printed, they expect a response from the system in about 2 seconds otherwise they think that the scan or label request or text-box entry has not taken and needs to be repeated, with sometimes duplicate materials being entered or packed or duplicate labels being printed.

With the old Access PC desktop program, they got immediate feedback because they were running on a dedicated PC that could give them an immediate response. But with a web-browser, the entered data item is sent to the server, without updating the displayed page until a response is received from the server.

At the server end, the data item (such as from a barcode scan) is received and then checked against the BellHawk database for data collection and operational correctness before a response page is sent back to the browser. This response page may be the same page as the browser was displaying but updated as a result of the scan or a page with an error message.

While BellHawk instructs the browser to turn the data entry box yellow until a response is received, when it turns green again, practical experience indicates that most users will ignore this and simply reenter the data again or make the label request again if they don’t see a response within a couple of seconds.

With today’s high-speed Internet and high-speed multi-core processors we can achieve response times of under 2 seconds on a server dedicated to running BellHawk. But, if BellHawk is run on a virtual server under control of a hypervisor, this will not work. The reason is that each virtual server gets up to one second of processing time before relinquishing control to the next virtual server. If there are five virtual servers, sharing one physical computer, for example, then there could be a four second delay before starting to process a barcode scan or label request. The result is a randomly occurring but totally unacceptable delay for people doing data entry into BellHawk.

This requirement for a dedicated server computer then brings up the issue of cost. A dedicated server for BellHawk is expensive, especially when you add in the cost of air-conditioning, security, backup power generators, and the like. This is in contrast to the Windows Server used in the old Access based systems, which often was a PC with the Windows Server operating system (O/S) loaded instead of the Windows Workstation O/S. For this reason, a number of our clients use a web-browser based BellHawk server (essentially a PC with Windows Server O/S loaded) located directly in their plant or warehouse, rather than at a remote data center.

One question I often get asked is “Don’t you run into the same problem with a large number of data entry users sharing a single BellHawk server?” The answer is that you do, with a very large number of users, but in most plants or warehouses barcode scanning is very sporadic so that collisions for server time are rare, as the processing time for each data entry value is typically less than one second. Also, if you use a multi-core processor for BellHawk, it can handle several scan or label printing requests in parallel.

To help mitigate this problem, when we have more than one geographically separated site, such as multiple manufacturing plants with an attached warehouses or multiple distribution centers, at different sites, we typically use a separate BellHawk system for each. This distributes the process load over multiple computers. It also helps solve another problem, which is people becoming confused if they see data for multiple sites in the same system. As a result, we recommend using a separate BellHawk system for each geographic site.

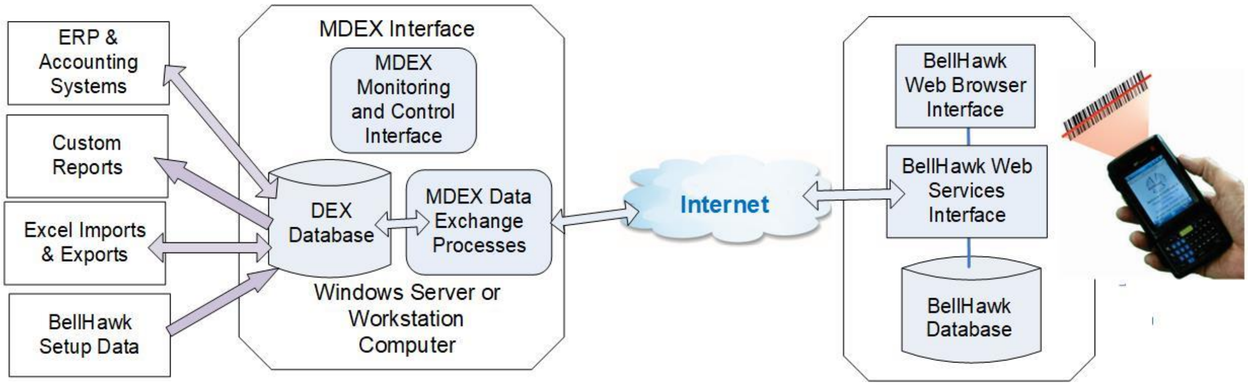
There are exceptions, of course, such as for field-service or building sites all served from a single warehouse or manufacturing plant or for warehouse-on-wheels delivery applications in a geographic region.

Another set of issues that we had to solve was how to enable users to configure the data collection screens to collect custom data and how to enable them to produce their own reports.

We solved the data collection issue in a similar manner to that used for barcode labels, in that we enable BellHawk systems administrators to import expert systems rules, for what data to collect, in the form of Excel spreadsheets. While this does not provide all the flexibility that modifying the old Access/VB code did, it mostly avoids the need for custom code development, which is a big advantage. And KnarrTek does offer a service to customize the BellHawk software, when needed for specific complex cases.

Report generation is a multi-faceted problem that we first ran into with our BellHawk Online service. While we could have a dozen or more users on line doing barcode scanning (with rapid response) the barcode scanning response time would slow to a crawl as soon as someone ran a large report or Excel export out of BellHawk, which was totally unacceptable.

The solution we chose to this was to have a mirrored SQL database (DEX) against which users could run reports as well as to use to exchange data with other systems:

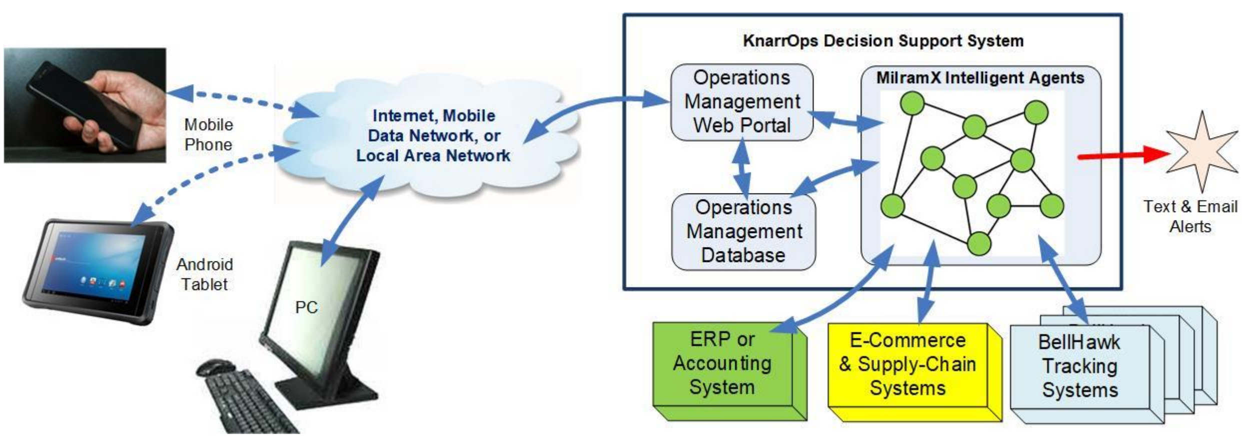


Tables in the DEX database, which is a SQL Server database, is updated whenever a change is made in the BellHawk database. Also changes to special tables in the DEX database result in data such as purchase orders, work orders, and ship orders being automatically updated in the BellHawk database.

The data exchange with BellHawk is done incrementally and is throttled so as not to interfere with barcode data collection and label print requests. Also, the DEX database is organized so as to be user friendly for generating reports, as opposed to the BellHawk database which makes extensive use of indirection and stored procedures for rapid processing of barcode scan data.

The MDEX interface is based on KnarrTek's MilramX intelligent-agent enterprise integration software and, as such, runs best on a multi-threaded Windows sever located in an organization's own data center. MDEX can be run on a virtual server as there is no requirement for rapid user response.

Where there are multiple BellHawk tracking systems in use, then the data for all these sites can be aggregated into one enterprise-wide Operations Tracking and Management databank using the KnarrOps software. KanrrOps is best run on a multi-threaded shared server at an organization's data center, where it can be accessed for reporting and viewing through a single-logon operations tracking and management web-portal.



One final issue that has arisen with BellHawk is a limitation in the speed of barcode scanning due to the time taken to access the BellHawk SQL database. We have observed some organizations using a separate computer, shared between many applications, to handle all their SQL database requests. While SQL Server Standard does a great job of making use of parallel processor threads to handle multiple requests in parallel, we have found that the free SQL Server Express, which is limited to a single thread, running on the same processor as BellHawk, can be much quicker in response for BellHawk scan checking than a heavily loaded, shared standard SQL Server process running on a much more powerful multi-threaded processor.

What we have observed is that BellHawk servers with local "disk" drives give much better response time, even when using single-threaded SQL Server Express, especially when the newest generation of high-speed M2 SSD chips are used for storage. But, of course, SQL Server standard running on a dedicated multi-threaded processor will give the best performance of all with BellHawk but the additional cost may not be justified for many applications.

Transition to the Edge

Our transition to Edge computing started about 3 years ago, when we observed that organizations using our TAG label printing software were:

1. Having trouble successfully installing the BellHawk TAG software and especially BarTender Automation on their own Windows Workstations.
2. Had problems with label printing suddenly stopping when Microsoft installed an upgrade and did a reboot in the middle of a production shift (often third shift).
3. Had problems with label printing when someone used the workstation as a general-purpose PC to run some other program, such as to watch a video.
4. Unable to provide KnarrTek with remote access support to assist clients in setting up printer drivers or modifying label formats.

To solve these problems, we now ship the BellHawk TAG and BarTender Automation software to clients, pre-installed on a small ruggedized Windows IIOT Enterprise based computer, complete with remote maintenance software, ready to plug into their local area network. KnarrTek can then remotely assist clients to setup label formats and printer drivers, ready for label printing without needing any changes to the organizations firewall or other security settings.

We now have three years of results with using these small Edge-based industrial computers as print servers and have found that, provided that they are plugged into a UPS (uninterruptable power supply), that they can run 24x7 for years on end without going down.

As a result of this experience, and the recent availability of low-cost 2, 4 and 8 thread ruggedized mid-range Windows IIOT enterprise computers, we are now offering BellHawk shipped pre-installed on a Windows IIOT computer.

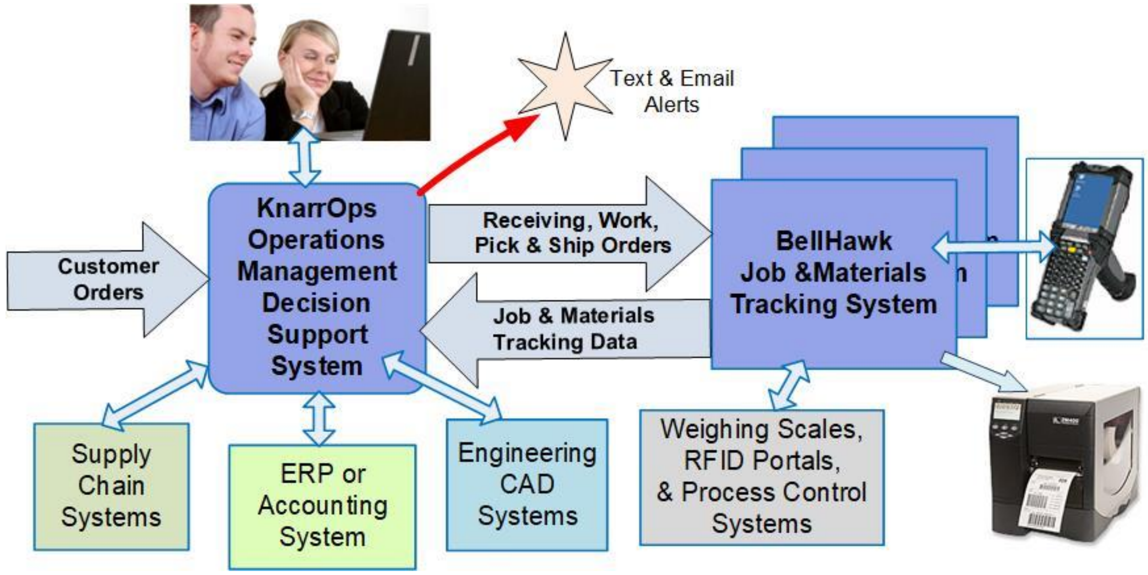
These BellHawk Servers are intended to be plugged into a site's local area network and to track manufacturing and warehousing operations at that site. With appropriate setup, however, users outside the site can still access the server over the Internet to remotely capture and view data, thus giving the best of both worlds.

By having a dedicated BellHawk server, with most data entry taking place locally over the plant or warehouse LAN, barcode scanning speed is maximized while still retaining the advantages of not having to install software on any of the data collection devices. And, if a local label print server is used, barcode label printing speed is also maximized.

The BellHawk server computer can also be remotely maintained over the Internet by KnarrTek, just like any other Cloud based server, eliminating the need for any on-site IT support.

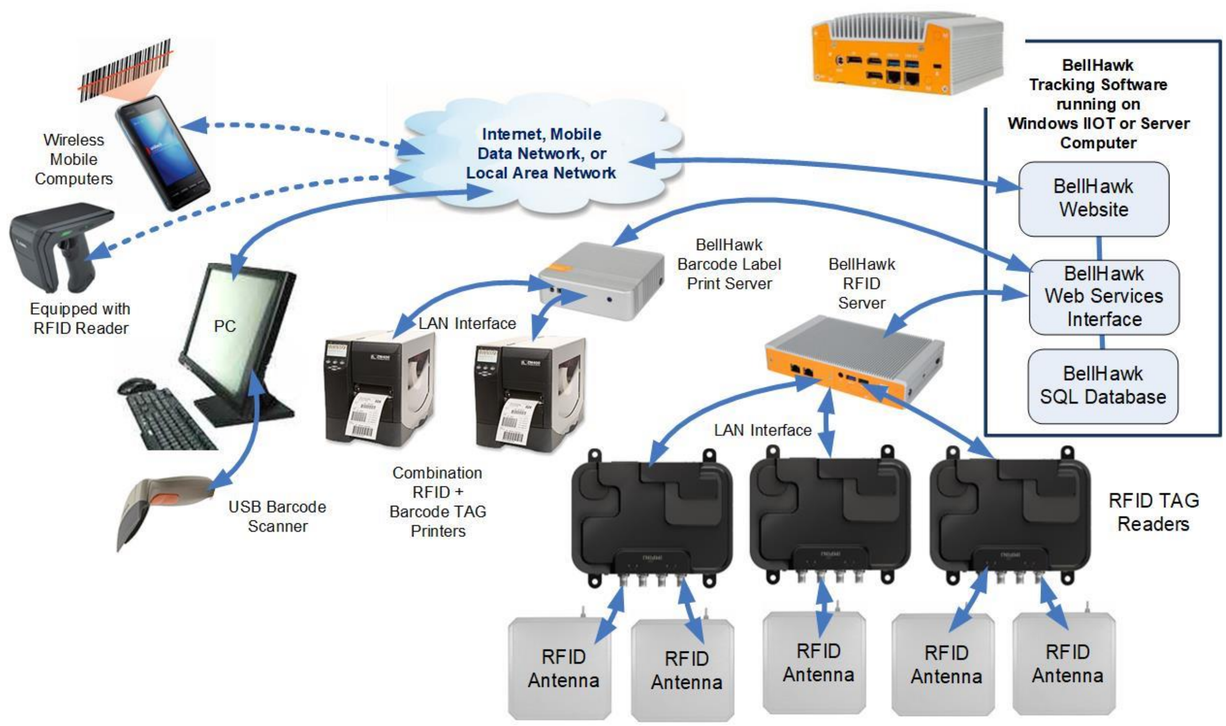
If the Internet goes down then data collection can continue, until Internet service is restored, as the BellHawk server is now local and not thousands of miles away.

Multiple BellHawk systems, at different sites, can be linked together so that they can be run as a coordinated group, using the KnarrOps Operations Management Decision Support System.



RFID Integration

KnarrTek also uses Windows IIOT Enterprise based RFID servers to integrate RFID tag readers into BellHawk. These RFID servers can communicate with BellHawk running on a local IIOT server as well as when BellHawk is running at a remote data center.



Commentary

BellHawk is now at Version 8 and has all the features, and many more, that were in the old V5 client-server-based system. It has all the advantages of V5 with none of its disadvantages.

With V8, KnarrTek is supporting both the Cloud-based and Edge-based versions of BellHawk, with a preference for Edge-based, as this usually results in superior performance. It is also supporting the IIOT based barcode label print server, the RFID server, and the weighing scale appliance (not described here) for use with both the Cloud and Edge based versions of BellHawk.

The big take-away from all this is that, except for software testing and pilot installations, BellHawk needs to run on a dedicated physical computer, with a dedicated database. The lowest cost way to now achieve this is to use a dedicated Windows IIOT based ruggedized industrial computer to host BellHawk.

Author

This white paper was written by Dr. Peter Green, who serves as the Technical Director of KnarrTek Inc. Dr Green obtained his BSC (Hons) in Electrical Engineering and his Ph.D. Degrees in Electronics and Computer Science from Leeds University in England. Subsequently Dr. Green was a senior member of technical staff at Massachusetts Institute of Technology and a Professor of Computer Engineering at Worcester Polytechnic Institute.

Dr Green is a Systems Architect who is an expert in using real-time artificial intelligence methods to implement real-time Inventory Tracking and Operations Management systems for Industrial Organizations. He has led the implementation of over 100 such systems over the past decade. Dr Green also led the team which developed the BellHawk job and materials tracking software, the MilramX intelligent information integration software platform, and the KnarrOps EDS software platform.

For further discussion, or to send comments, please contact peter.green@KnarrTek.com.

Copyright