



White Paper

Belden Horizon™ demonstrates the benefits of a Container & Microservices cloud architecture

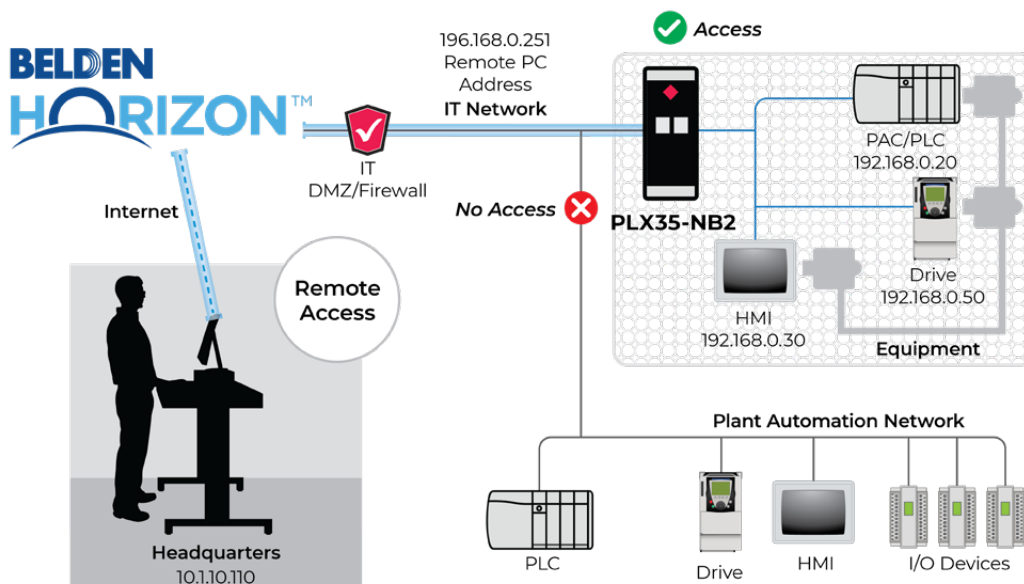
By Keith Blodorn
Business Directory, ProSoft Technology

ProSoft Technology has delivered valuable products and services helping industrial customers connect automation equipment across different networks for nearly thirty years. Today, as its customers are looking to benefit from the Industrial Internet of Things, we have a secure, cloud-native platform called Belden Horizon™ to help customers do more. Belden Horizon provides some unique advantages because of the Container and Microservices cloud architecture that it's built on. This paper describes the Container and Microservices architecture and illustrates how ProSoft customers benefit from it.

What are Containers and Microservices?

“Containers” and “Microservices” are terms used to describe an approach to developing applications for use in a cloud environment. The term Container brings to mind a shipping container – a standardized box used in shipping to make it easier to move products overseas, through ports and across railways. In cloud application development, a Container is a standardized set of services that support the application running inside the container and provide connection to the cloud host environment. Microservices are the applications that run inside the Containers. As opposed to writing a single, monolithic program to deliver all user functions, developers build Microservices that perform a very specific task. Various Microservices come together to form what appears to the user as a complete cloud-based service. Yet, since each Microservice and the Container it resides in is fully self-sufficient, these functions are not dependent on each other.

This concept is a departure from the predominant software development approach of the recent past. For many years, software developers built full programs that would install on a computer and run in a specific computer operating system. All of the functions of the program exist within the code, and the entire program is installed together. The program then depends on the operating system to provide the required connections to the world (input devices like



the keyboard, output devices like the monitor, network connections, etc.). In addition, all of the functions within the program depend on each other. This approach to programming worked well as long as there would always be a fairly complex host to run the program – the computer operating system.

As more computing functions are moved to cloud hosted systems, this paradigm is no longer the ideal way to develop computer programs. Many applications have been ported to the cloud simply by running a virtual machine with a specific operating system, and installing an existing PC- or server-based program to that virtual machine. This is a very inefficient approach since many aspects of the operating system are not needed by the program. Further, since all of the functions of the program are interdependent, a failure in any part of the program can crash the entire service. Booting up a new VM with the service can take minutes. Reliability is improved by adding “failover” servers, which are entire copies of the virtual machine running side by side.

Containers and Microservices provide a better way to build software for cloud deployment. This approach begins with the compartmentalization of program functions into Microservices. A Microservice is just a specific function of the program. Think of the payment service on an online retail site. The overall service may have many such Microservices providing different functions. The other key element of this approach is the Container. The Container is a standardized interface between the Microservice and the rest of the world, similar to how the operating system provided the interface for the monolithic program in the old paradigm. Containers offer quite a few advantages in the cloud hosted environment. They are much less resource-intensive compared to virtual machines and full operating systems, so it's easy to deploy multiple instances of a particular Microservice to handle traffic and provide better reliability. Containers “spin up” in less than a second, so failure recovery is extremely fast. Making functions fully independent from one another also allows the development team to use the best programming language for each function, rather than choosing one language for all functions. If one function is most easily deployed in python, and another function in C++, each function can be developed in the optimal language for that function and deployed in its own Container.

Think of the old approach of monolithic programs running on virtual machines as a warehouse. The warehouse, like the virtual machine, is designed to house any manner of contents. It is quite large and time-consuming to build, and the contents inside are all dependent on the integrity of the warehouse – and can even be affected by problems with other contents. In addition, each warehouse is built with different configurations – aisles, shelving, doorways – so moving contents from one warehouse to another will require changing how the contents are stored. When things are busy, the warehouse may be full, while at other times, it will be nearly empty. Yet it takes up the same amount of space either way.

By contrast, shipping containers are really very small, modular warehouses. Each one has standard dimensions, handling provisions, and configuration. Containers hold all kinds of different contents, yet it is very easy to manage many containers by stacking them on ships or in shipyards. Individual containers take up much less space than a warehouse, and it's relatively easy to add containers when more contents need to be stored. In busy times, the shipyard may be full of containers stacked in rows. When it's not busy, far fewer containers are needed. If anything happens to a container, only the contents inside that container are affected, and a replacement can be ready quickly.

Just as the advantages of shipping containers make logistics operations more efficient, cloud-native container technologies make cloud-based programs easier to develop, deploy, and operate.

How the Container and Microservice Architecture Benefits ProSoft Customers

Belden Horizon allows customers to securely access industrial automation devices – such as PLCs and HMIs – remotely from anywhere in the world. Secure remote access allows system integrators, machine builders, and large end users to troubleshoot problems with their systems more quickly, to gather data from machines or plants around the world, and plant operators to manage process equipment spread globally. The unique advantages of the Container and Microservice architecture includes enhanced service reliability, improved security, and superior ease of use.

For manufacturing customers, reliability is always a critical attribute. Secure remote access services are no different. When a machine is down, engineers need to access the machine control system and begin troubleshooting immediately. Belden Horizon users enjoy highly reliable service in part because the Container/Microservice approach allows the service to run multiple simultaneous sessions with minimal consumption of cloud computing resources. In fact, there are always at least three of every service running! There is no need for the “fail-over” servers that old VM-based systems used for reliability. Even if all of the current instances of a service were to suddenly stop, a new instance can be started up in under a second. This means that when an engineer needs to access a remote machine, Belden Horizon will be ready to make the connection.

As more industrial processes connect to the Internet, cybersecurity is an important aspect of connectivity solutions. The Container and Microservice architecture helps Belden Horizon deliver a higher level of security for users. First, Belden Horizon is a fully cloud-native solution that requires virtually no user-installed software. This eliminates a significant attack surface that past cloud connectivity solutions opened up. User-installed software is vulnerable to tactics like “watering hole attacks,” where hackers embed sniffer code inside the software download and use that to gain access to equipment. Also, software must be kept up-to-date as vulnerabilities in the underlying software services

are discovered and patched. Belden Horizon requires only a browser and an OS-native VPN client to work, so these threats are greatly reduced.

Second, the Container and Microservice architecture enables Belden Horizon developers to build a very robust and secure service using the latest state-of-the-art cloud development tools. Containerized functions prevent potentially cascading issues, since each function operates autonomously in its own Container. Finally, Containers provide the basic supporting services that a piece of code needs to run. This means there are fewer components that might be vulnerable to attack, compared with a full VM and operating system.

The most noticeable advantage of the choice to use Containers and Microservices is in the ease of use. Belden Horizon functions were built using the best programming language and supporting services for each function, which contributes to the simple user experience of the service. For creating VPN connections to remote equipment, Belden Horizon offers EasyBridge™ technology, which forms a Layer 2 connection between the user's PC and the remote PLC network. Containerized services running in the Belden Horizon service handle the complexity of network routing, so the user can connect to the remote network just as if they plugged an Ethernet cable into the remote Ethernet switch.

A Platform for the Industrial Internet of Things

Perhaps the most exciting aspect of Belden Horizon and its modular architecture is the promise of things to come. The Industrial Internet of Things (IIoT) is a hot topic these days, and for good reason. As new technologies develop to help manufacturers cut costs, improve productivity, and deliver products faster, Belden Horizon users will have a powerful secure platform to connect industrial data to high-value cloud services. We already help customers around the world get data from machines speaking one protocol to machines that speak a different protocol. We've now built the foundation to help customers take those machines and deliver the value the IIoT is promising.

Learn more about Belden Horizon.



Keith Blodorn has worked in the Industrial Automation industry for more than 20 years.

Support and expertise for your projects

ProSoft provides a variety of resources and services for wireless applications from initial system design to site surveys to launch support and configuration support.

Contact Us

Asia Pacific Regional Office

Phone: +603.2242.2020
asiapc@prosoft-technology.com

Regional Tech Support
support.ap@prosoft-technology.com

North America Regional Office

Phone: +1.661.716.5100
info@prosoft-technology.com

Regional Tech Support
support@prosoft-technology.com

Europe/M. East/Africa Regional Office

Phone: +33.(0)5.34.36.87.20
europe@prosoft-technology.com

Regional Tech Support
support.emea@prosoft-technology.com

Latin America Regional Office

Phone: +52.222.264.1814
latinam@prosoft-technology.com

Regional Tech Support
support.la@prosoft-technology.com

