

Crunching Enterprise Containers for Industrial IoT

In a world trending toward agile software practices and cloud-native development, containerization has become a handy tool in the enterprise development toolbox. Technologies like Docker allow developers to bundle applications, libraries, configuration files, and other utilities into a neat little package. Because everything is “containerized” into this package, the application can move from one computing environment to another and execute just the same.

In other words, containers make software about as portable as it can get.

But they are also an attractive option for non-enterprise use cases, such as industrial IoT edge devices. As these packages and their contents are partitioned from each other and the rest of the system, they can be individually updated without impacting the rest of the software stack. In an industrial IoT context this enables:

- Workload consolidation by allowing multiple applications to run on a single piece of hardware
- Lightning-fast system patches, updates, and new service delivery
- Easier codebase management and version control, as out-of-date application containers can be updated with precision across an entire deployment

Still, there are reasons that packaged-up software hasn't been adopted more readily in operational environments, namely due to determinism and functional safety concerns.

But before tackling those issues, it's necessary to understand how packaged-up code architectures work and their potential role in operational environments.

Containers Versus Virtualization and Resource Constraints

As shown in **Figure 1**, containers are a lot like virtualization technology, but not the same. They function as application packages on top of a specialized operating system (OS), as opposed to providing a virtual environment in which an OS runs. Because they house everything needed to run the application, you can think of containers as virtualization at the OS level.

“When you run a true container environment, everything's running on the same operating system,” said Ron Breault, Director of Business Development & Marketing for Industrial Solutions at Wind River. “You're not virtualizing your operating system anymore. You may have multiple processes running that are given different privileges so that it looks like they have their own system, but they're all sharing the same kernel.”

But except for select versions of Linux and a few specialized options, most operating systems do not support containers. This is because a specialized orchestration engine is required that, similar to a hypervisor, helps supply containers with various resources like drivers, networking stacks, security, failure recovery, and others (**Figure 2**).

While critical to container operation, the problem with these orchestration engines is that they add significant overhead to the underlying OS (and by extension, the rest of the system). The impact of resource drain is substantial. Even though application containers are much smaller and boot more quickly than VMs, the management and orchestration infrastructure is a nonstarter for most embedded systems.

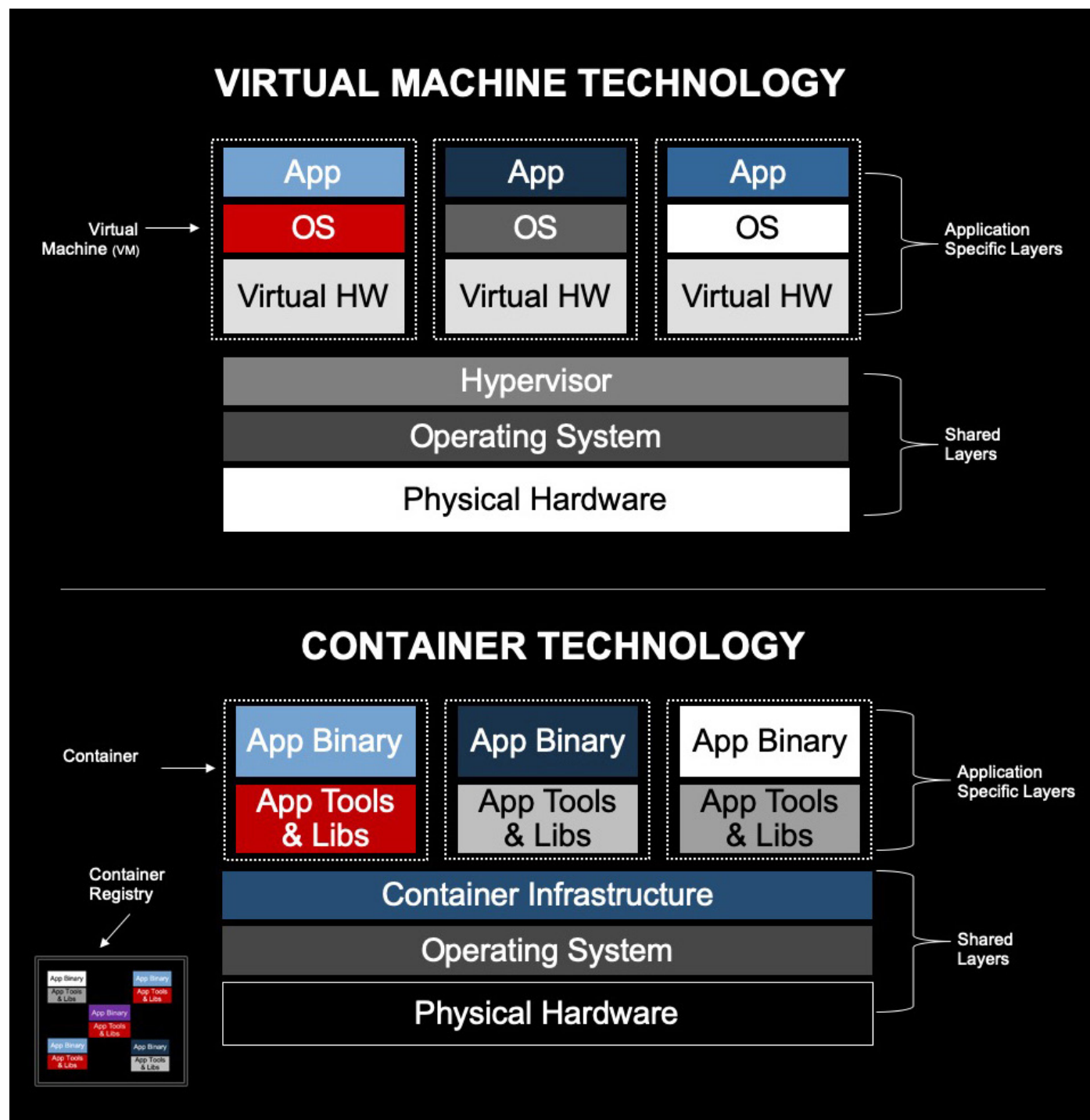


Figure 1. Virtualization provides an environment for launching additional operating systems, while containers package all of an application's dependencies into bundles on top of an OS. (Source: Wind River)

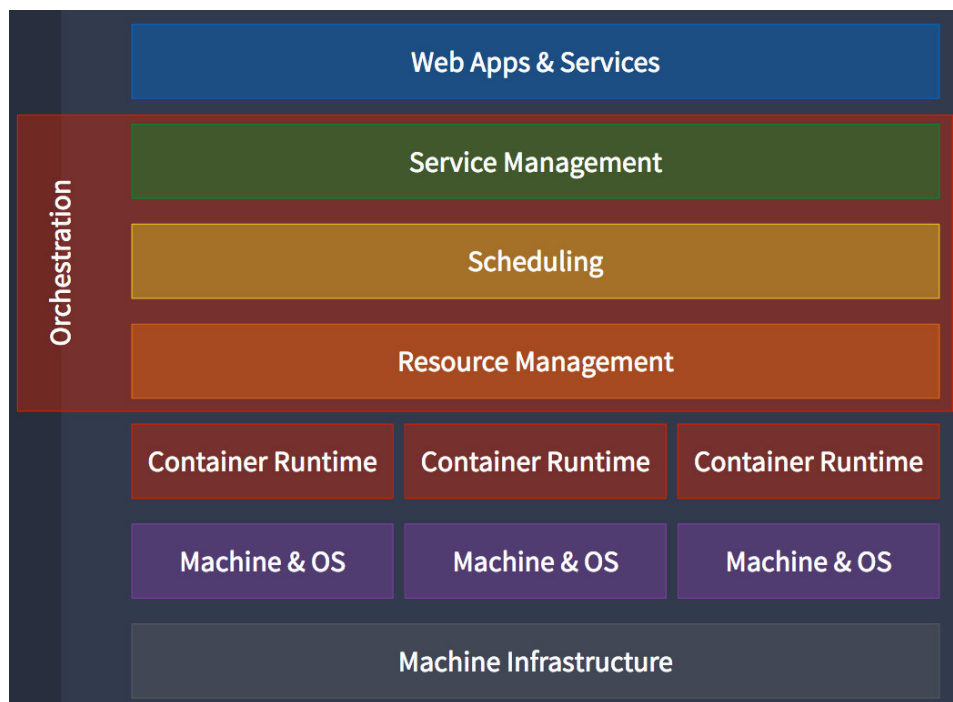


Figure 2. Container orchestration engines are similar to hypervisors in that they manage resources and services that a given container may need. (Source: [Devopedia](#))

For that reason, [Wind River](#) recently added the open-source OverC container technology into its latest version of Wind River Linux.

Crunching Containers for the Industrial IoT Edge

OverC is a much lighter-weight version of popular container technologies like Docker and Kubernetes, yet provides most of the same advantages. And because OverC is compliant with the Linux Foundation's Open Container Initiative (OCI), it is compatible with other container technologies so that applications and images can move seamlessly among the platforms.

Just as important, integration of OverC into Wind River Linux opens up a world of possibilities for mixed-criticality systems. As Wind River Linux is natively supported by the Wind River Helix Virtualization Platform, engineers can leverage the precision updating and software lifecycle management capabilities of containers alongside safety-critical OSs in a workload-consolidated system (**Figure 3**).

The Helix Virtualization Platform is based on the Wind River Hypervisor, which can be certified to safety standards such as ISO 26262, DO-178, IEC 61508, and so on. In practice, this means that developers could spin up a containerized version of Wind River Linux in one VM and a real-time control task in another, both robustly partitioned by the hypervisor.

"If you look at a manufacturing environment, this presents an opportunity to take different machine controllers and consolidate them onto VMs," said Jeff Kataoka, Senior Product Line Marketing Manager at Wind River. "Then in the container VM you could run the other data analytics and enterprise apps that might be needed. Now you can have it all on the same platform."

Outside of turning single-function appliances into multifunction platforms, the benefits of mixed-criticality containerization can be extended based on the use case. For example, organizations can reuse existing software investments by housing it into a secure VM. Or they can

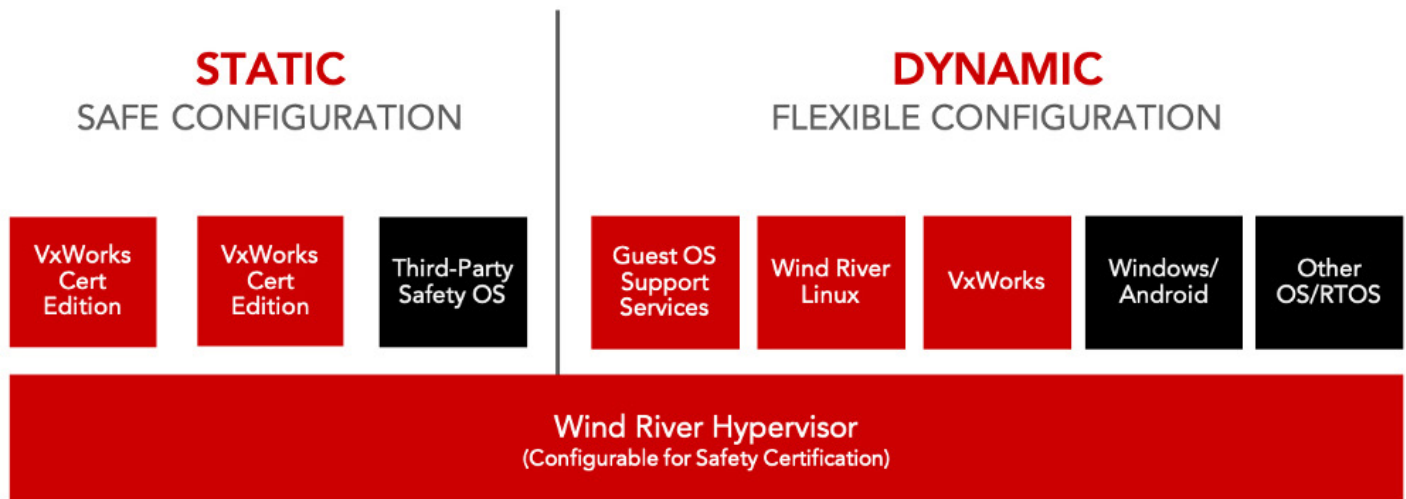


Figure 3. The Wind River Helix Virtualization Platform supports Wind River Linux and its recently added container features. (Source: [LinuxGizmos](#))

reduce certification costs by securely partitioning certified code from the rest of the system using the Wind River Hypervisor.

What's more, the Helix Virtualization Platform can run on multicore processors with virtualization support, ranging from Intel Atom® or Intel® Core™ to Intel® Xeon®. The result is a scalable, flexible edge infrastructure that supports agile software deployment practices.

Cloud Native, at the Edge

Based on the sheer numbers of enterprise developers versus industrial engineers, it's likely that operational systems trend toward data center technologies.

Containers enable a write-once, deploy-anywhere approach, and also simplify remote software updates. In addition, along with virtualization, they help organizations reduce the amount and diversity of their hardware infrastructure, providing a direct return on investment.

Of course, there will always be a need for embedded software. Indeed, it is the embedded software that controls equipment that most enterprise applications are analyzing anyway. This intersection is the mixed-criticality middle ground between data center infrastructure and operational technology.

With containers, hypervisors, and the Wind River Helix Virtualization Platform, we've reached that middle ground.