

# Top Five Considerations for Deploying Next-Gen Appliances

---

# Introduction

Today's software application buyers are demanding more options than ever when it comes to application delivery methods. Delivery requirements include applications that will run on-premise, in a virtualized environment, in the cloud, as a hardware appliance and more. These delivery methods are in addition to the range of choices customers already expect in terms of platform, operating system, database, etc. In response, original equipment manufacturers (OEMs) and independent software vendors (ISVs) have to provide multiple versions of an application to accommodate an expanding mix of environments.

Economically, this is a challenge. Rather than focus on providing business value with innovative features and functions, ISVs and OEMs may actually spend more time engineering the product for different platforms. One way that vendors have found to remedy this situation is to deliver the application as a pre-packaged, purpose-built hardware appliance.

A hardware appliance allows an IT department to essentially "set it and forget it." Once the box is installed in the data center and configured, it typically runs with little maintenance or upkeep required. And since it is self-contained, it doesn't have to "play nice" with the other system resources and components. This ease of use angle is attractive to customers and so it is enticing for vendors to deliver applications as hardware appliances.

But, here's the caution. While it may seem easy at first glance to deliver a hardware appliance, there are five critical considerations for deploying next-generation appliances:

- How do you harden the appliance operating system?
- How can you monitor the performance of the appliance stack?
- How will you upgrade and maintain the appliance over its lifetime?
- How can you ensure a repeatable, consistent build process?
- How should you support virtual and cloud deployments?

This paper will examine these questions and present a solution based on the partnership between rPath, Inc., and NEI and the intersection of their technology capabilities.

---

## A Hardened, Secure Linux Operating System

One of the biggest benefits of a hardware appliance is that it is self-contained — the operating system, the application and other components have already been installed, configured and tested to run well together. There are no dependencies on system components outside of the box. Even the hardware is optimized for the needs of the application to meet specific performance characteristics. But just because it is self-contained doesn't make it invulnerable. Any application that is connected to the network is susceptible to attacks from inside and outside the organization. The more lines of code there are, the more surface area — attack vectors in security language — is available for a malicious entity to exploit.

What ensures the security and performance of the appliance is a hardened operating system (OS) — one that has been stripped down to the bare minimum needed to run the application. This hardening process could be attempted manually from a commercial OS, with a detailed analysis of the application, its OS needs and the resulting dependencies. But this is a painful and time-consuming process with little potential for repeatability.

In the classic packaging model, large pieces, such as utilities or desktop tools — or even an entire monolithic OS — are unnecessarily included in the OS. If you need only a library in your package, you often get the entire application that it belongs to because they can't be separated. This results in a bloated OS, with a larger footprint that is more open to security threats.

NEI recognizes that this is unacceptable. Not only is a bloated OS a security risk, but it is also a performance drag. Why optimize the hardware if the software is going to slow things down? That's why NEI uses rPath rBuilder® to ensure a hardened Linux system that contains Just Enough Operating System (JeOS). JeOS is not a generic, one-size-fits-all OS. Rather, it refers to a customized OS that precisely fits the needs of a particular application.

rBuilder allows NEI to essentially start with the Linux kernel and build up from there, rather than start with a monolithic OS that needs to be reduced. Starting with the smallest possible footprint and adding packages, rBuilder eliminates OS bloat with its ability to distinguish smaller components, such as libraries, documents and executables. Fine-grained dependency resolution ensures that the final image is the smallest possible footprint — in many cases up to 80% of its original size. In addition to reducing the security vulnerability of the appliance, an added benefit of this simplicity is that less can go wrong, since you're delivering fewer components. Testing and problem resolution become much easier when there are fewer dependency trails to follow.

### A hard OS is good to find

At the heart of rBuilder is the repository where the components, applications, libraries and builds reside. In order to build an appliance, the application package is loaded into the repository. rBuilder analyzes the package and breaks it down into its individual components. Think of a component as containing the files that serve a particular functional role in the application.

rBuilder then analyzes each of these components, automatically determining any component dependencies. The software component and its related dependencies are then stored and versioned in the rBuilder repository.

When it is time to construct the hardware appliance, all that is needed is to tell rBuilder to construct a system image containing the desired application. rBuilder uses the detailed component level dependencies to ensure that all the needed software components — and none that aren't needed — are included in the image. The result is a minimal, hardened Linux OS that reduces vulnerability while increasing application efficiency.

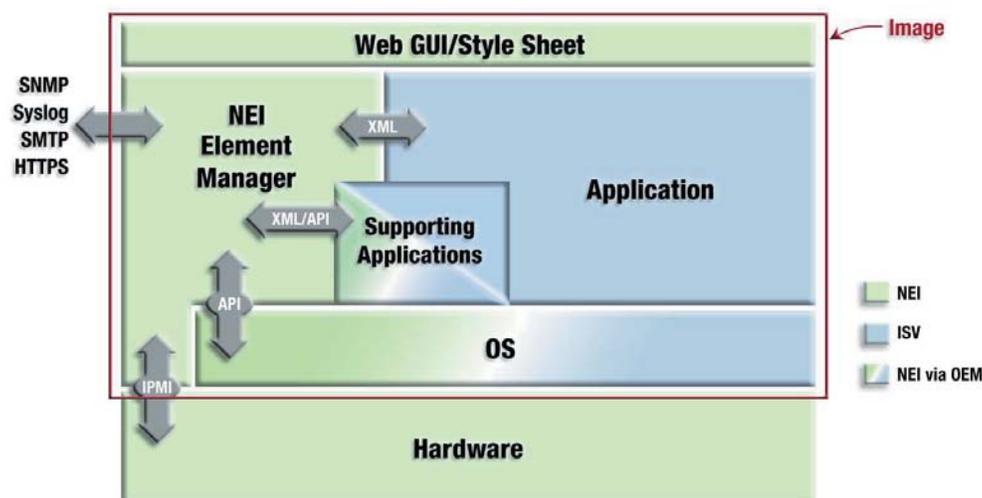


Figure 1  
NEI's Appliance Model

NEI's appliance model seamlessly integrates all core hardware and software components of the appliance. The company's Element Manager helps ensure higher availability by routinely auditing and reporting the operational condition of critical components, including CPUs, power supplies, storage arrays and disk drives, as well as the OS and application itself.

## Performance Monitoring

Optimal appliance performance and uptime result from both hardware and software operating properly and at peak efficiency. Unfortunately, any number of things can go wrong and the best way to reduce downtime is to prevent it from happening at all. NEI Element Manager integrates with the software components to monitor the health of the appliance and proactively alerts you of potential problems.

NEI Element Manager routinely audits the operational condition of core appliance hardware elements, including CPUs, power supplies and disk drives, and software events, such as crashes, runtime errors or system boot failures. In the event of hardware or software malfunctions, NEI Element Manager automatically generates a maintenance request or administrator notification.

NEI Element Manager can also send alerts to a third-party network monitoring system. An alarm handler uses pre-set rules to determine which events are actionable and may need intervention. Its browser-based console puts powerful management controls and tools at the administrator's fingertips.

Addressing problems before they affect performance or cause downtime is key to maintaining productivity and user satisfaction. NEI Element Manager gives you the edge when it comes to correcting small issues before they become big ones.

## Appliance Upgrade and Maintenance

The days of viewing hardware appliances as disposable commodities are over. Whereas before, it might have been acceptable to swap out a perfectly good box for another one with a newer version of the application, it's now seen as economically and ecologically wasteful. Instead, customers want an easy way to extend the life of their hardware without sacrificing new features and functionality.

To do that, ISVs need to deliver maintenance releases and upgrades to deployed boxes without disrupting operations. NEI Update Services automate the process and management of delivering updates, patches and other upgrades to boxes deployed in the field, including the OS and all related applications. NEI relies on the version control capabilities in rBuilder to create change sets which become the patches that are installed to deployed appliances.

### No news is good news

NEI Element Manager works by interfacing with the main application and supporting applications through XML or APIs. It also extracts information from OS logs and integrates that data into its analysis of the health of the entire solution. NEI Element Manager determines whether information in the logs is relevant and needs to be tracked.

To connect with the hardware in the appliance, NEI Element Manager can also use the intelligent platform management interface (IPMI). It can be configured as to which events to track, such as power failure, disk failure, etc. When fault conditions occur, NEI Element Manager automatically generates an alarm event for investigation or maintenance requests. Its health and alarm manager helps centralize alarm management and reduce unnecessary support requests.

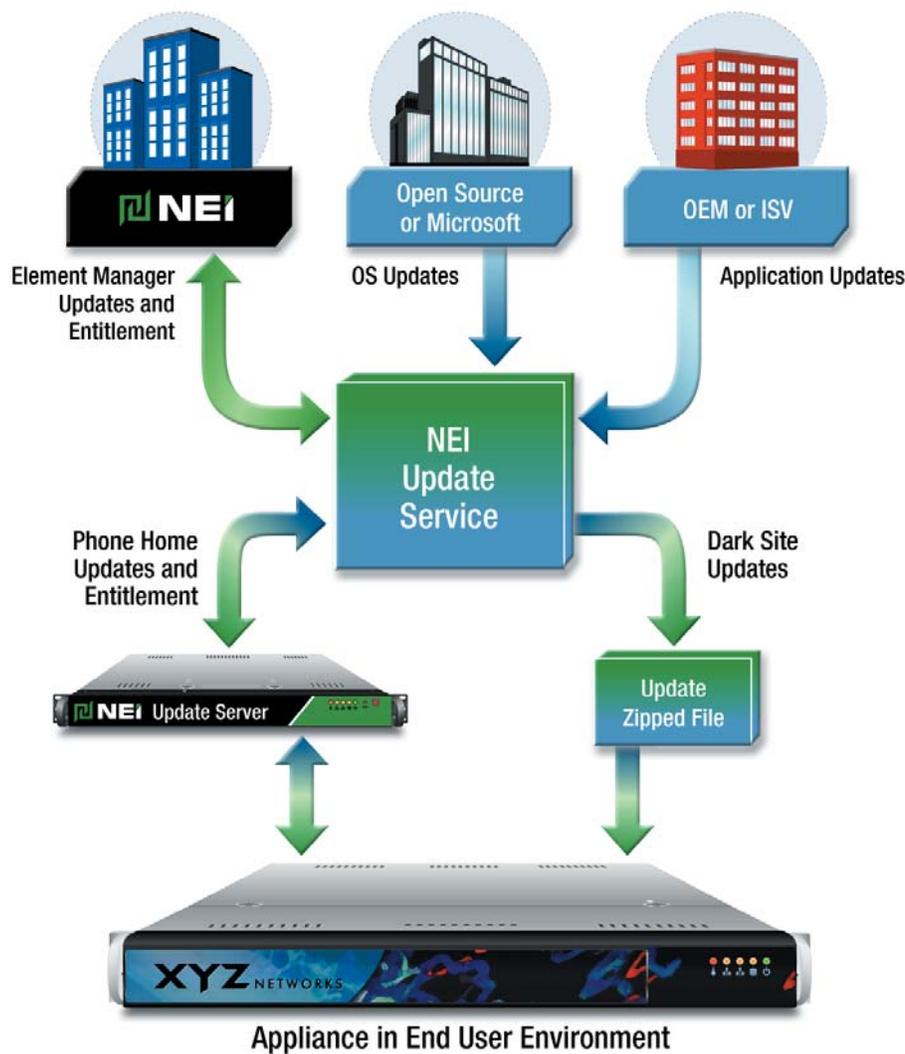


Figure 2  
NEI's Update Service

Fully automated updates to the operating system, application and software support components are applied remotely and securely

NEI Update Services are included with every appliance built by NEI using rBuilder. The service provides a centralized control for all the operations involved with extending the useful life of a hardware appliance, such as:

- Secure requests for the latest updates so that nothing is pushed to the appliance without permission
- Delivery of encrypted manifests, patches and OS updates
- Complete system backup and rollback capabilities
- Phone home and dark site support

As an added benefit, updates and patches are delivered incrementally to the appliance, so that only components that have changed are affected. This reduces the time and bandwidth required to perform an update, and also minimizes disruptions.

## Repeatable Build Process

In a manufacturing facility, it is expected that processes are automated, consistent and repeated from one item to the next. This allows hundreds or thousands of units to be built so that one is indistinguishable from another. A uniform product makes sense to maximize economies of scale, ensure quality and facilitate support. If a product specification changes, then the factory tooling is altered to produce that change. The change is not added manually to finished products.

Apply this same approach to building software, and you'll understand why NEI uses rBuilder for version control. With rBuilder, the entire system, together with all of its components and dependencies, are stored and managed in a centralized repository which provides consistent control over the lifecycle of the deployed system.

When the appliance is built the same way, every time, numerous problems that arise from inconsistency are eliminated. The application remains the same from development through test, QA and deployment. Similarly, every change to the application can be tracked, audited and rolled back if necessary. Nothing is left to chance or human error so that every appliance is indistinguishable from the next.

## Support for Virtualization

Up until now, we've been discussing building, upgrading and maintaining images for hardware appliances. But many ISVs and OEMs recognize the need to support virtual environments and even cloud computing platforms. Without specialized tools, building images to support popular image formats, such as VMware, Citrix Xen and Amazon Elastic Compute Cloud (EC2) machine images, can be time-consuming and onerous.

One significant advantage to using the NEI/rPath solution is that once an appliance is defined in rBuilder, it can be output to a number of popular formats without any additional engineering effort. This particular feature provides hardware appliance vendors with the flexibility they need to produce these formats.

### Repeatable Process. Repeatable Process...

With rBuilder, automation and repeatable processes drive the creation of software components and system images. At the lowest level, software components are retrieved from sources, compiled, and packaged via automated scripts. Sources and the resulting binaries are versioned and stored in the repository, allowing a direct connection to be maintained between the applications and the code base that created them.

Likewise, system images are the results of scripts that define only the desired applications – all of the dependent components need not be defined. During the image creation step, rBuilder automatically determines the required components, their specific versions, and stores all of this information in the repository along with the image.

It is this all-inclusive approach to both automating the build process and to versioning and storing the results that gives rBuilder its unique advantage.

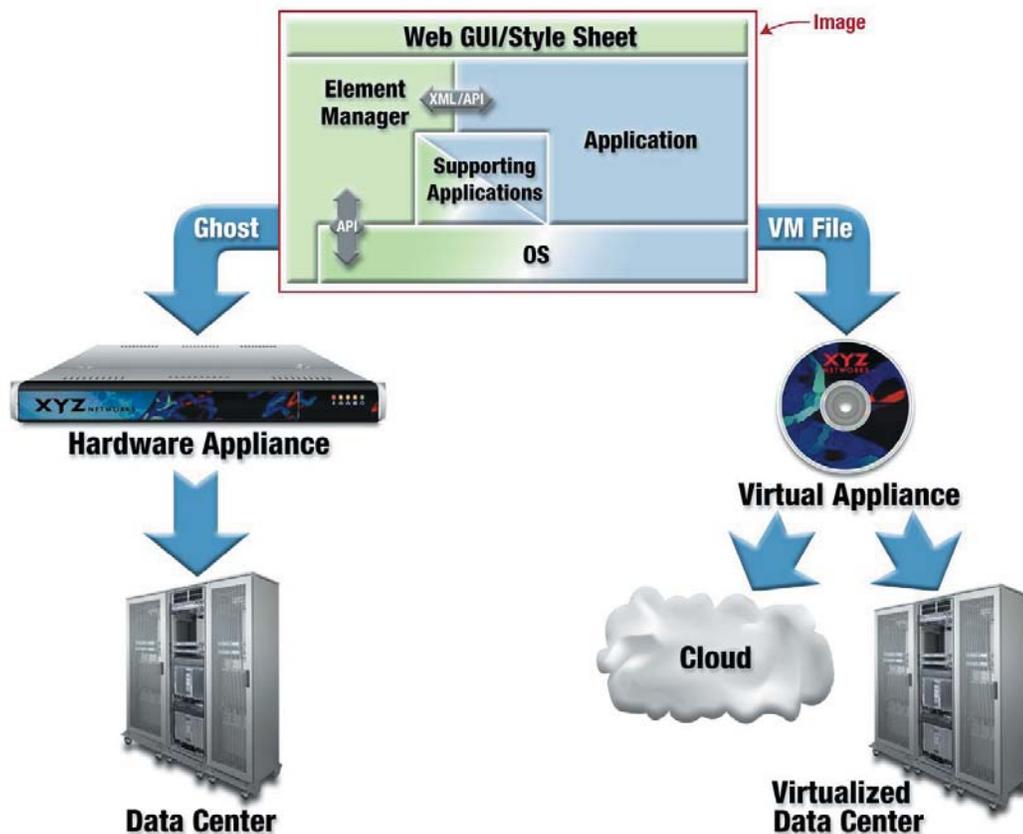


Figure 3  
NEI's Virtual Product Deployment

Similar to hardware deployments, virtual deployments enjoy the same rich features and capabilities. When properly created they are easy to deploy, manage, support and update.

Many organizations use virtualized or cloud-based environments for development, test and QA activities. Others use virtual or cloud images for demonstration purposes, since it may be easier than carrying a hardware appliance on sales calls! No matter what the reason for using an alternative environment, once you've created an appliance definition with the NEI/rPath solution, a virtual or cloud image is just a click away.

### A Winning Partnership

While NEI and rPath have partnered to create a solution that manages the entire appliance lifecycle, the real winning partnership is the one you'll enjoy with NEI. With an experienced technical team, NEI works with you every step of the way to ensure your appliance meets the highest standards of quality and performance.

As you consider delivering your application as a hardware appliance, we hope you'll recognize the importance and value of lifecycle management – from development through deployment, upgrade and maintenance. When your appliance is built with the NEI/rPath solution, you get a secure, hardened operating system and a total appliance solution.

## About NEI

NEI is a leading provider of application platforms, appliances and services for software developers, OEMs and service providers worldwide. NEI enables customers to more effectively deploy, manage and support application platforms and appliances using its comprehensive capabilities, including solution design, integration control, global logistics, smart services, technical support and maintenance. Founded in 1997, NEI is headquartered in Canton, Massachusetts and trades on the NASDAQ exchange under the symbol NENG. For more information about NEI's products and services, visit [www.nei.com](http://www.nei.com).

## Contact Us

For more information on the full range of appliance-related hardware, software and services NEI can provide to your company, please contact us by telephone +1 (781) 332-1295 or by email at [sales@nei.com](mailto:sales@nei.com).