WIND™

Embedded Linux

Prototype to Production

Abstract

Linux is the default environment for most software developers and is the most popular choice for embedded solutions. However, one of Linux's greatest strengths, and to some extent its biggest challenge, is that it comes in so many flavors and varieties—each well suited to a particular use case.

While everyone recognizes the strength of enterprise Linux distributions like Red Hat, SUSE, and Ubuntu for general-purpose IT use cases, they are generally not suitable for embedded uses. Compared to general computing, embedded solutions have a lot more constraints and have higher reliability and security requirements, lower resource availability, and support needs that often span five to 10 years or more.

In response, many organizations attempt to fork an existing community distribution and create their own customized embedded Linux. This is often referred to as roll-your-own (RYO) Linux. While the initial costs of this approach are very low, the resources for support and maintenance spike in later years, as the expertise of the RYO must be maintained in-house.

Commercially supported embedded Linux is the primary alternative to RYO. Not only are the long-term support and maintenance costs much lower, but the technical, business, and legal risks of commercially supported embedded Linux are much lower as well. Commercial vendors can also provide full development life-cycle services, including platform, services, maintenance, and support, which increases productivity and reduces the overhead of maintaining your own, unique, embedded Linux distribution.

TABLE OF CONTENTS

WHICH LINUX?

Everything starts with the Linux kernel that is available from kernel.org. To build a full operating system for application development and deployment, additional packages are required. Knowing what you are going to build determines which packages are required for your distribution. There is not a one-size-fits-all. With this in mind, a Linux-based project could be classified into the categories discussed below.

Enterprise Linux

The general-purpose server and desktop Linux distributions from Red Hat, Ubuntu, and others are intended for well-resourced, multipurpose, and often multiuser use cases. As such, they are typically configured to support a wide range of devices with a one-size-fits-most mentality. Although the distributions are supported with available source code, supported customizations are usually limited to package installation and configuration files. This produces a solid user experience, "good-enough" reliability for general-purpose use, and an inexpensive support model with a three- to five-year horizon.

Embedded designs typically need much smaller footprints, greater performance in a constrained environment, higher reliability, longer support horizons, and much lower resource requirements than can be provided by enterprise Linux distributions.

Roll-Your-Own / Customized Linux

The roll-your-own approach involves forking and stripping down a community-based distribution. The thinking behind this model is that the developers can rely upon the existing community for support; however, the reality is that the community knows nothing about the actual fork that has been created and is thus unable to provide significant long-term support. Instead, dedicated engineers who understand this unique OS are needed to maintain and support it. That's not a big burden in the early years, but as time goes on, this burden grows significantly, and since everything is custom, there is little to no benefit from economies of scale to help support this project.

Every RYO Linux platform is effectively a unique and custom Linux distribution that creates heavy maintenance and support burdens in later years.

DO YOU REALLY WANT TO SUPPORT YOUR OWN UNIQUE LINUX DISTRIBUTION?

Newcomers to embedded Linux often don't understand that every RYO Linux platform created in-house is by definition a unique and custom Linux distribution. Embedded Linux requires compiling and linking on a host system for download to a target host.

The creation of all of the right puzzle pieces to make the OS run on an embedded target is effectively an in-house distribution of Linux. For this reason, embedded Linux is different from desktop and server Linux, where one typically uses a well-known distribution already packaged and ready to go. Companies deciding to roll their own Linux must understand that they are taking on the maintenance and support of a custom Linux distribution, which requires significant expertise and staff, **a commitment that dramatically increases over time.**

While community support for Linux abounds for current releases of the kernel and distribution releases, support quickly wanes as versions become outdated compared to the most current releases.

It's unrealistic to expect open source community support for an older version of the Linux kernel that, for example, has been built into a product and been on the market for several years. The community is typically focused on the leading edge of development, and older components (kernel, libraries, packages) are mostly left unmaintained, unsecured, and unsafe to use. As the years go by, companies must rely on internal specialized expertise to support their products. This goes well beyond the capabilities of most organizations' IT departments.

Beyond the heavy personnel costs involved, organizations must ask themselves if this is how they want this engineering talent tied up—in maintenance—or if their time would be better spent in product development. Indeed, there's a risk that companies might lose key engineering talent altogether if those employees are not enthusiastic about maintaining an old embedded distribution.

Managed Services with Wind River

Wind River® offers end-to-end Linux development products and services – from prototype to optimization, to deployment and beyond. Wind River Linux subscription service delivers Yocto Project-based source code for platform developers. It includes maintenance, long-term support, compliance artifacts, and security updates. All enhancements and fixes to the Yocto Project-based Linux source code are upstreamed for the entire community to leverage. Wind River also provides a premium service to help customers customize, manage, and optimize their Linux platform and applications. Wind River also provides a premium service that delivers a dedicated team to help customers customize their distribution, manage source code and binaries, support frozen branch distributions, and optimize their Linux platform and applications to meet specific market requirements.

Commercially Supported Embedded Linux

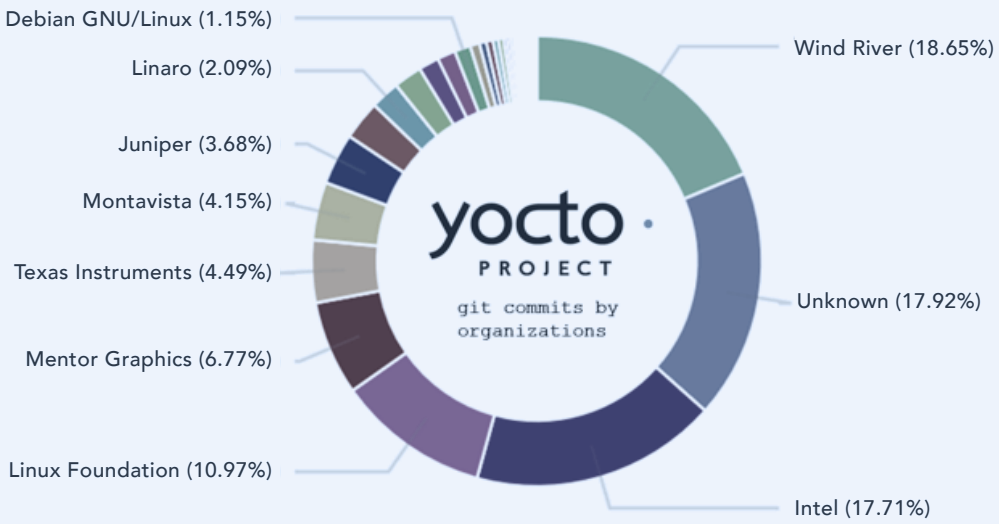
The third major category of Linux is a commercially supported embedded OS, such as Wind River Linux. The major advantages of this approach are very similar to those of enterprise Linux, but the solution is built with embedded use cases in mind. Instead of creating a massive one-size-fits-most distribution, the approach favored by most commercially supported embedded Linux vendors is to create a compact core able to support low-resource environments, very high reliability needs, and a build system that requires only local support for extensions.

This core framework is generally provided by the OpenEmbedded and Yocto Project communities, and enables *both* economies of scale in terms of support and highly customized builds that exactly fit project requirements.

The nature of embedded Linux is that, just as in an RYO situation, every deployment is a unique distribution. The difference with commercially supported embedded Linux is that these distributions are all sourced from the same code and carefully monitored, maintained, and patched by a consolidated team that is thus able to achieve significant economies of scale.

What is the Yocto Project?

The Yocto Project was created to standardize the platform for embedded Linux, thereby greatly increasing interoperability and creating an ecosystem of hardware and software vendors around a common set of standards and tools. The ecosystem also provides support and expertise toward emerging standards, such as those being created today for the Internet of Things. The Yocto Project is freely available and provides a standardized embedded Linux build system for production use.



The Yocto Project is a highly active and healthy open source community for embedded Linux.

The OpenEmbedded Project

OpenEmbedded serves as the build framework for Wind River, the Yocto Project, and many other embedded Linux systems. OpenEmbedded offers a best-in-class cross-compile environment and allows developers to create a complete Linux distribution for embedded systems.

GET TO MARKET FASTER

While it's true that the greatest savings from commercially supported Linux accrue from more efficient support and maintenance over the long haul, it also offers many opportunities to speed up the development process and get product to market faster.

Board Support Packages

Board support packages (BSPs) provide massive time savings as well as a big leap forward in reliability and quality compared to building your own drivers and hardware support package.

What is a board support package?

A board support package (BSP) is the layer of software containing hardware-specific drivers and other routines that allow Linux (or a real-time operating system like VxWorks) to function in a particular hardware environment. BSPs are customizable, allowing the user to specify which drivers and routines should be included in the build based on their selection of hardware and software options.

A BSP should be customized not just to the correct architecture/OS combination but down to the exact system on a chip (SoC) and board configuration used.

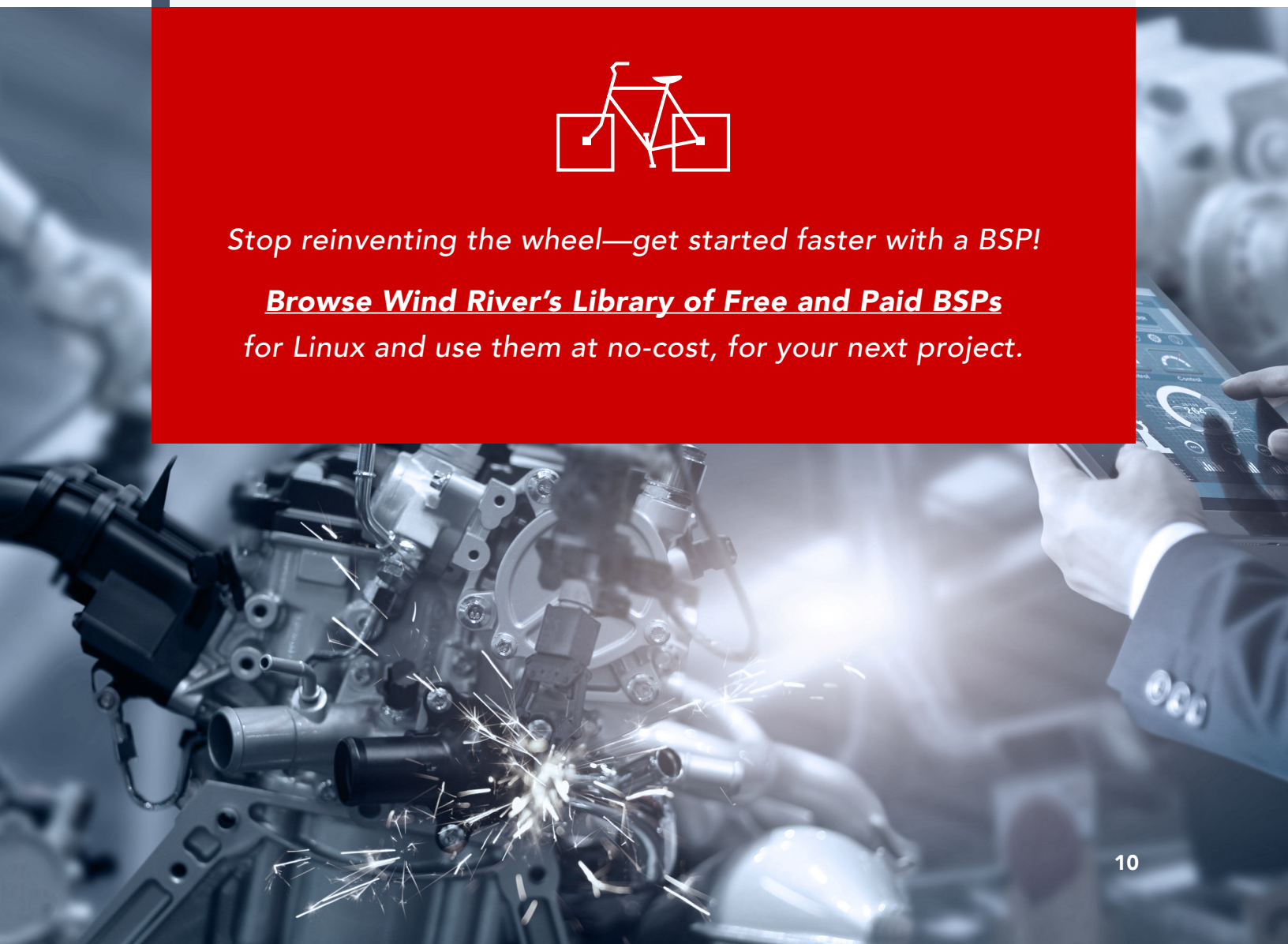
Of course, since everything in an embedded environment should always be as optimized as possible, the BSP also includes a customized version of Linux (or other OS if applicable). Additionally, your BSP vendor can provide customized test suites, developer assistance from experts in your chosen hardware, and other professional services that can save your team time and unhappy surprises.

In other words, BSPs provide a polished development and production environment perfectly tailored to your use case.

Why Use a Commercially Supported BSP?

The primary advantages of working with a board support package supplied by a commercial vendor versus building your own are speed, expertise, and economies of scale. In addition, commercial BSPs are supported and maintained. If you run into a problem, you can get help.

Wind River, for example, has thousands of BSPs ready to go covering virtually every popular processor architecture, including ARM, MIPS, PowerPC, and x86. They work with almost a hundred different hardware platforms and even if they don't have an existing BSP for a given hardware configuration, they can use a similar one as a template and quickly create a highly secure, stable, and efficient operating platform.



Stop reinventing the wheel—get started faster with a BSP!

Browse Wind River's Library of Free and Paid BSPs
for Linux and use them at no-cost, for your next project.

Cloud-Native Architecture and Container Technology

Cloud-native architectures and containers are widely deployed in enterprise IT environments but have been largely missing in action for embedded systems. However, the benefits of containers are equally desirable for embedded systems:

- Code reusability
- Efficient maintenance
- Platform independence
- Optimized resource utilization

Until recently, leveraging cloud-native architectures and containers for embedded systems meant rolling your own custom solution based on platforms that were optimized for data centers… with all of the support and maintenance nightmares that implies. Most current container frameworks lack the right design or support for mission-critical industries that typically employ devices with extremely long lifecycles. These existing container technologies and platforms, like those in enterprise Linux, are often bloated or require updates too frequently to run effectively on these embedded systems. So most embedded systems still use traditional "bare metal" physical architectures.

However, the latest version of Wind River Linux provides substantial support for teams looking to bring cloud-native architectures to embedded systems.Wind River Linux now includes pre-integrated components from the Cloud Native Computing Foundation (CNCF) configured to deliver a fully functional solution for embedded systems such as edge appliances.

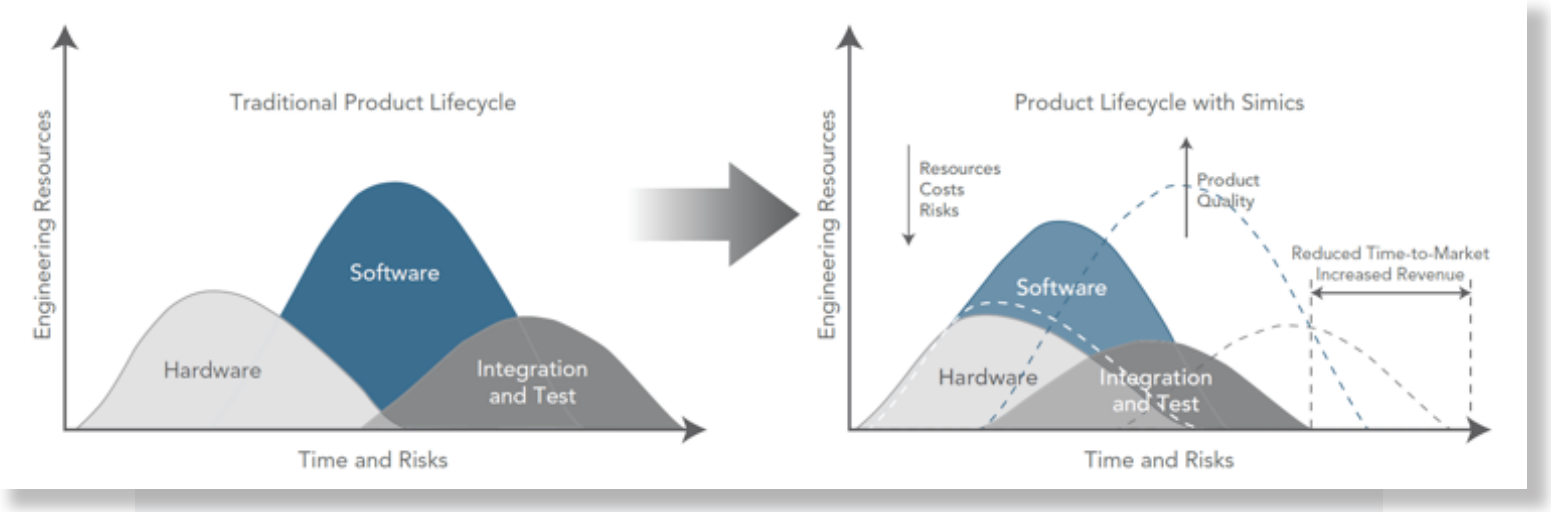
Wind River Linux now includes pre-built containers and support for Docker, Kubernetes, etc.

Development and Build Tools

Commercial Linux is accompanied by value-added development tools that go well beyond what is available from (mainly command line) open source tools. In addition to the familiar coding, testing, and debugging capabilities, modern commercial integrated development environments provide tools for building the Linux and device image, system performance and behavioral analysis, memory profiling, and tracing.

Full System Simulation

An example of an extremely useful category of software tools is the full-system simulation—the simulation of not just the embedded target processor, but also the system-on-chip (SoC) features, board-level hardware, and even interconnected targets in a complete system of subsystems. Developers, with their commercial development tools, can deploy and test on a simulated full system. This innovation greatly improves the support for iterative and agile development processes, but also removes the hardware supply bottleneck that plagues most embedded development projects.



"The ability to shorten our development period by 15% was a remarkable result of using Wind River Linux. This led to a considerable reduction in development costs."

—Takahide Inoue, Sharp Corporation

LONG-TERM SAVINGS

Support and Maintenance

In the long run, commercial offerings that provide a proven embedded Linux with support and maintenance are generally significantly less expensive than maintaining a roll-your-own (RYO) Linux solution in-house. Linux is large and complicated, and requires significant effort by your engineers to provide support, patches, and security vulnerability management over time.

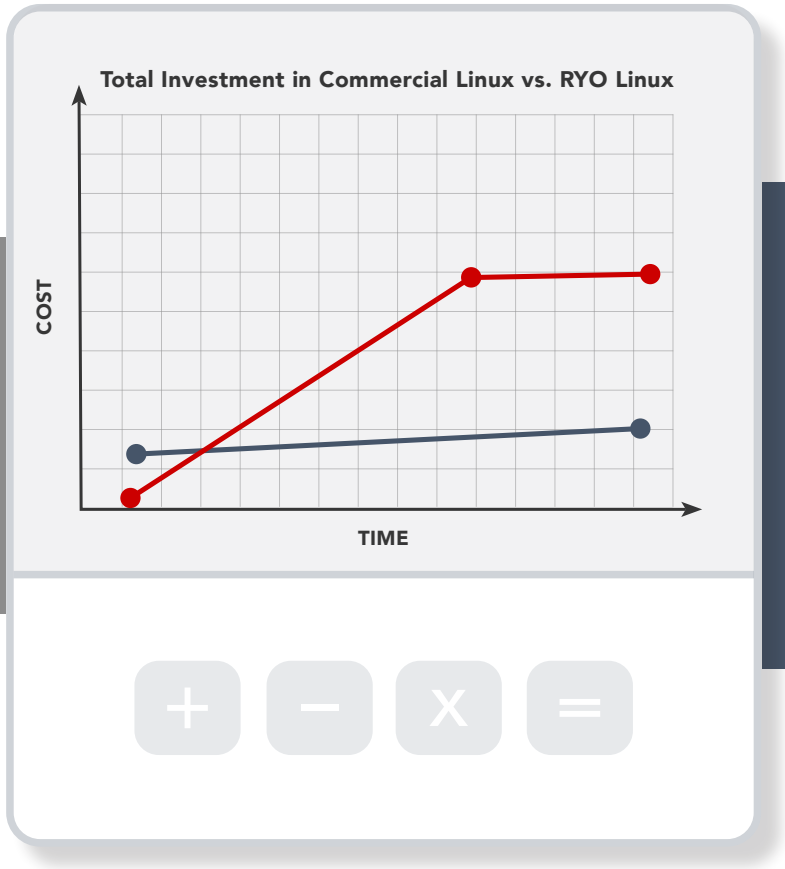
While it's difficult to make broad generalizations about the cost of creating and maintaining your own embedded Linux, that cost is always considerable and often wildly underestimated by organizations new to the field. For instance, the kind of investment required for a Linux-based operating system that would be suitable for a device used in telecommunications is shown in Table 1.

	Costs
Initial release (porting, testing, hardening, quality assurance)	$600K
Yearly maintenance	$200K
Yearly security vulnerabilities assessment and resolution	$250K
Yearly export classification and open source licensing compliance	$50K
5-year total costs for RYO Linux	~$3M

Table 1 shows a typical cost for maintaining a Linux-based telecommunications device for five years.

The primary variables in the long-term cost of maintaining your system are:

- The type of embedded device
- The life span of your devices
- Security requirements
- Virtualization
- Application footprint
- Internet connectivity
- High availability or fault tolerance requirements
- Global or purely local deployments
- Free open source software / open source policy requirements
- If you will be updating devices that are in the field

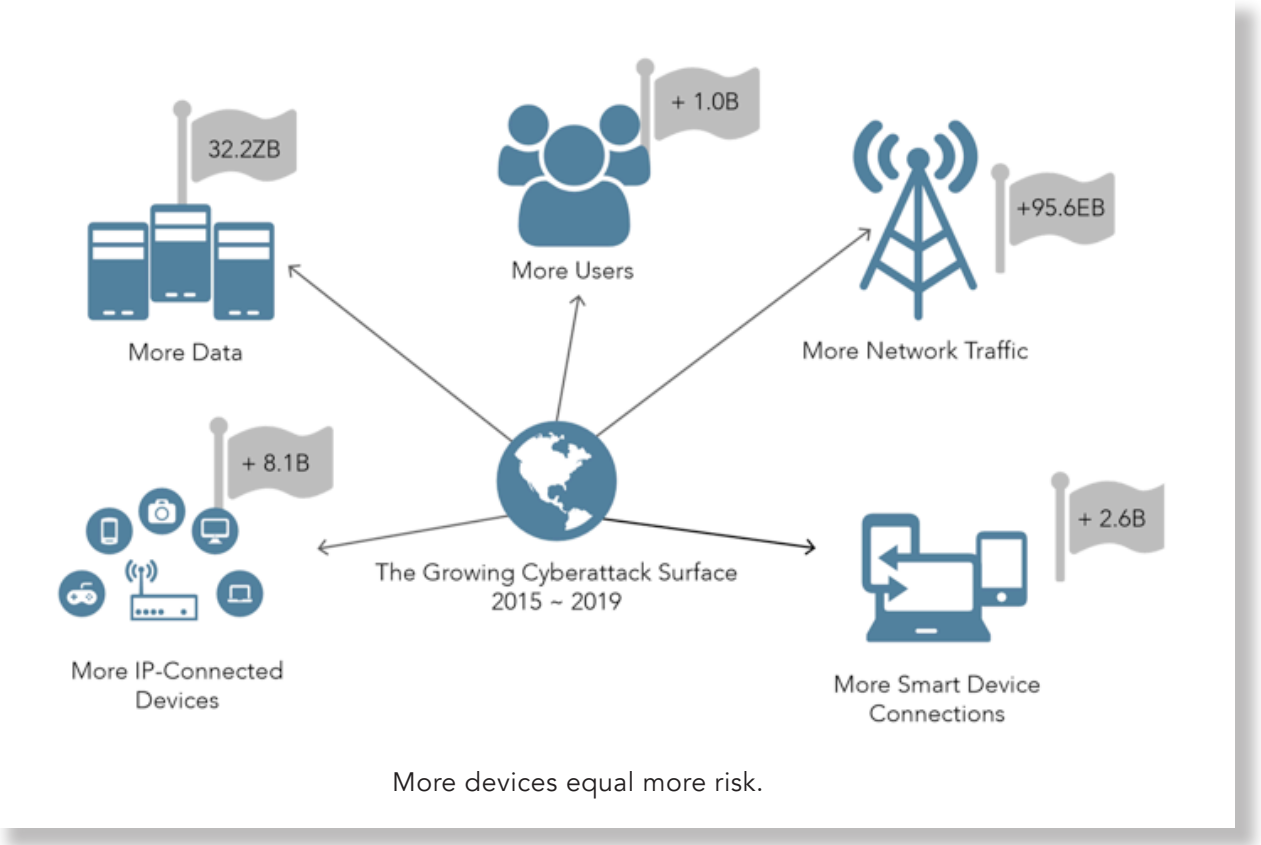


Calculate the total costs of ownership for your specific use case for both roll-your-own and commercially supported embedded Linux using our online cost calculator.

Better/Cheaper Security

With ever-increasing numbers of interconnected IoT devices being deployed, Linux software vulnerabilities have become more widespread than ever. Taking responsibility for identifying vulnerabilities and making the necessary updates to mitigate threats is often beyond the capacity of device developers and manufacturers.

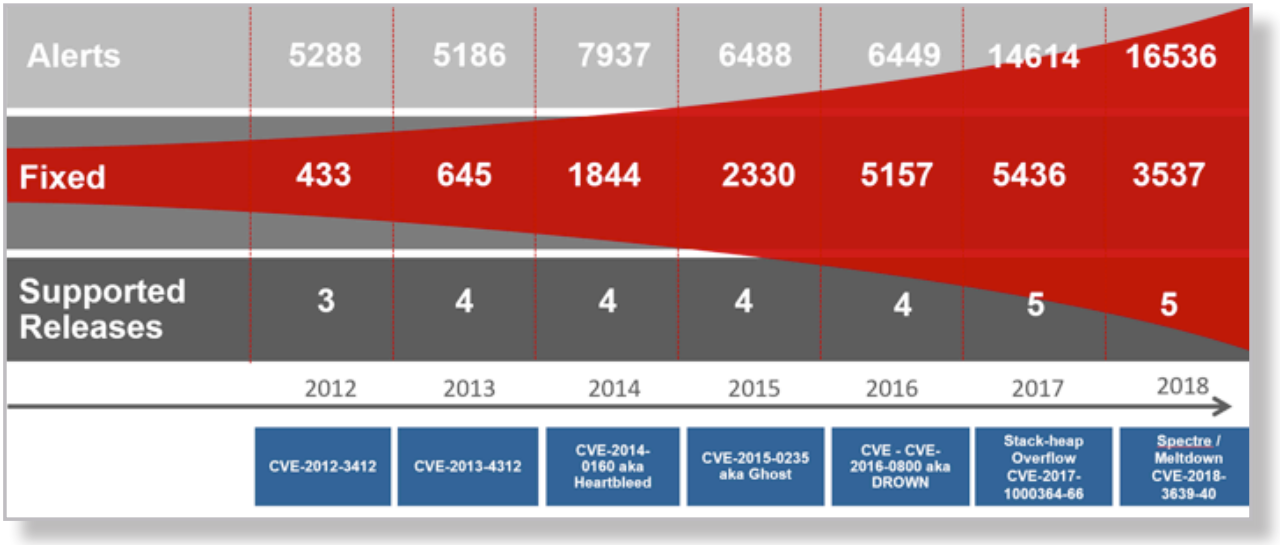
Gone are the days of "fire and forget" device deployment. Virtually every device made these days is designed for interconnectivity with something, which makes them susceptible to security vulnerabilities. The reality is that connected devices are very likely becoming more vulnerable with every reported exploit.



More devices equal more risk.

Managing and mitigating threats is essential for the protection of end users but requires a level of engagement that is beyond the scope of most IoT solution developers, device manufacturers, and system operators.

THE COST OF SECURITY MONITORING



Proactive security monitoring and patching is needed to stay in front of critical vulnerabilities.

Wondering how many security threats your device will likely face before end of life? ***Check out this interactive security threat calculator.***

Commercial Linux vendors provide regular product updates and maintenance and security patches. Due to economies of scale, these vendors can devote the necessary resources to stay on top of Linux kernel and security updates with regular patch schedules and convenient delivery methods. This level of expertise and proactivity would be very hard to achieve for most embedded device manufacturers.

BEWARE OF COMPLIANCE ISSUES

Of course, there are more than labor and technical costs to consider with embedded Linux; it's also important to understand the legal implications of using Linux in embedded systems and the risks associated with open source licensing, IP and export compliance, and so forth.

Licensing Headaches

Creating your device with a Linux run-time system as part of its software is equivalent to distribution under many of the open source licenses used in Linux, including the GNU Public License (GPL). There are on the order of 20 million lines of code for Linux and associated open source tools—a massive code base with a multitude of licenses for organizations to trip over if they're not diligent.

> *Unlike using Linux on a desktop or server, shipping a product with Linux is legally considered to be redistribution, which opens you up to more licensing requirements.*

With redistribution comes the responsibility to make sure your company is complying with the license requirements, such as providing free access to the source code for the open source portions of your product, including any tools that might ship with the product.

Often there are clauses in the licenses about derivative works that can include kernel modules, libraries or tools, modification of existing code in Linux, statically linking to open-source libraries, and other stipulations. It's critical that embedded device manufacturers catalog the licenses of the software they are using, understand the level of risk associated with the license, and be prepared to fulfill the obligations associated with each of the licenses. Unfortunately, many companies don't treat this aspect of Linux seriously and open themselves up to needless liability, impact on brand, and cost to fix problems after the fact.

"50% of companies will face challenges due to lack of free open source software (FOSS) policy and management."

—Gartner, FOSS Analysis, November 2011

Commercial solutions make it possible to reduce risk, time, and cost. Commercially supported Linux delivers full licensing compliance and reporting. Commercial vendors are experienced with Linux licensing and can aid in the adoption of open source into your code base mix.

Wind River and Open Source

Wind River is best known for its VxWorks real-time operating system, but it has also been a leading provider of embedded Linux solutions for more than 15 years. Wind River is a founding member, advisory board member, and the largest single contributor to the Yocto Project. Since the Yocto Project's unveiling in 2010, Wind River has been a Yocto Project Advisory Board Gold member, contributing key technology to the project and collectively investing more than 89,000 lines of code over the past four years. Wind River actively upstreams and contributes fixes and enhancements back into the project while building Yocto Project support into the next generation of the Wind River platform.

Wind River has dedicated five full-time employees and additional engineers working part time on the Yocto Project, collectively investing almost 240 months over the past five years.

We believe that encouraging and building on standards makes embedded Linux stronger in the long run.

Export Compliance and Encryption Disclosure

Preparing products for international export adds another layer of compliance complexity to the documentation of open source software. In addition to the necessary license compliance requirements, export compliance largely centers on the disclosure of cryptography software, which presents security concerns in many countries.

This is an additional reason software suppliers, application developers, and device manufacturers need to have formal processes in place for tracking open source software (OSS). When it comes to documenting OSS in general and cryptography in particular, many technology companies experience a disconnect between the engineering and export teams. Export teams typically expect engineering to know everything in the code base so they can properly report on the cryptography used in a product. If the product has a large number of OSS components and the engineers did not actually write the code, however, they may not have a clear understanding of the cryptography inside. But export disclosures rely on accurate information from the engineering team, so organizations need to improve the quality of their cryptography discovery in OSS.

When there are hundreds or even thousands of OSS components within a product, a manual search is not practical. Some type of automated tool is needed, but automation alone is likely to yield some false positives, which then have to be reviewed manually. The most efficient solution—the one Wind River employs—is a combination of automation and encryption expertise. First, a tool is used to search the code for encryption, and then a designated team trained in encryption technology analyzes the findings to weed out false positives. A report detailing the levels and types of cryptography found is generated and added to the compliance envelope. The export team can then more accurately determine which instances of cryptography need to be reported based on the requirements of the country.

ISO 9001 Certification

Wind River Linux development and maintenance processes have been certified to the ISO 9001:2015 quality management system standard. The certification covers the design, development, integration, verification, validation, and maintenance of open source technologies when packaged into Wind River Linux.

In addition, Wind River builds quality into its products through its commitment and conformance to the Wind River corporate product development life cycle and the underlying ISO 9001:2015 engineering and non-engineering processes it references.

CONCLUSION

Commercial embedded Linux offers a clear return on investment versus RYO, in-house developed, and maintained Linux. Not only is the total cost of ownership lower, but the technical, business, and legal risks of commercial embedded Linux are much lower. The ability of the commercial embedded Linux vendor to supply training, services, maintenance, security updates, and support greatly increases productivity and also reduces the overhead of maintaining your own, unique embedded Linux distribution.

Success in a competitive market means concentrating on what you do best—building great products—and avoiding the risk and expense of platform development that adds to neither your feature set nor your bottom line.

Carrier-Grade Linux? Check.

Need carrier-grade reliability? Wind River Linux offers carrier-grade capabilities, which deliver 99.9999% availability, clustering, security, virtual routing and forwarding (VRF), test suites, and more.

ABOUT WIND RIVER

Wind River is a global leader in delivering software for the intelligent edge. The company's technology has been powering the safest, most secure devices in the world since 1981 and is found in more than 2 billion products. Wind River offers a comprehensive portfolio supported by world-class global professional services and support and a broad partner ecosystem. Wind River software and expertise are accelerating digital transformation of critical infrastructure systems that demand the highest levels of safety, security, and reliability.

TO LEARN MORE, VISIT WWW.WINDRIVER.COM.

According to the VDC Research Group, Wind River Linux commands more than 52.1% of the commercial embedded Linux market.

Custom Content & Frozen Branch Maintenance

Wind River can maintain your customized Wind River Linux platforms or a specified standard product throughout your product's life span.

MONITORING FOR SECURITY ISSUES AND PATCHES

BACK PORTING PATCHES FROM LATEST WIND RIVER LINUX RELEASE

FIXING BUGS IN OPEN SOURCE PACKAGES

TESTING

CUSTOMIZED PLATFORM DEVELOPMENT