THE NEED FOR OPEN SOURCE AT THE EDGE

RTInsights.com

WIND™

CONTENTS

CHAPTER 1

What Is the Intelligent Edge?

There is a perfect storm driving the adoption of intelligent edge applications—the availability of new technologies and the need for businesses to embrace digital transformation and real-time operations. According to Gartner, while only about 10 percent of enterprise-generated data has been created and processed outside a traditional data center or cloud, by 2022, that's expected to increase to 75 percent. And the location of the data being generated is the edge. Fortunately, the growth in data at the edge and the need to analyze it there is happening at a time when there are significant developments in the open source software community.

From a technology standpoint, the intelligent edge is possible due to the advent of hardened compute solutions that are small enough to be deployed in edge environments. New connectivity services such as 5G provide a conduit to connect edge devices into existing systems. Open source software provides the needed operating system (OS) and other features for edge devices to run. And the growing use of sophisticated analytics solutions and artificial intelligence (AI) makes it easier to extract information and insights from data at its source of generation on the edge. To put the growing importance of intelligent edge into perspective, note that Gartner estimates that 80 percent of all data will be processed at the edge by 2025.

Enabling solutions across the intelligent edge is intrinsically tied to business success as the world digitally transforms.

Intelligent edge systems can streamline the analytics process when there are hundreds or thousands of devices. Rather than sending all the data from every device to a data center or cloud computing center, edge devices can make critical decisions locally and choose when and how to push that data to the cloud—or decide that such a push is not necessary at all. Besides enabling more responsive systems, bandwidth requirements are drastically reduced. Given the cost to transmit data, companies can realize significant savings. Additionally, less data is sent between points, lowering the risk of it being stolen or spied on.

Enabling the use of data at the edge is supported by the proliferation and adaptability of open source software, which supports the vast array of communication protocols used at the edge and the edge's footprint.

There also are situations in which bringing intelligence to the edge is the only solution—specifically, when there is not enough time to send data from its point of generation to a data center or cloud facility for analysis, and then send the information back in time to act. Autonomous systems are the perfect example. A Mars rover or an autonomous car both need to make instantaneous decisions based on the analysis of video images and positional data. Transmitting the data to a central location to spot and avoid a crevasse or object in the road will not do. A number of established organizations and standards bodies promote and support openness and interoperability for these solutions, all rooted in open source principles.

Similarly, an intelligent edge system can keep operating in times when no connectivity is available. For example, a logistics application could continue to work when a truck enters a tunnel or is driving through a region with no cellular data service.

By bringing responsiveness and innovation where it is needed, intelligent edge devices are fueling digital transformation. These capabilities make the intelligent edge ideal for many application areas across many industries, including healthcare, manufacturing, retail, smart cities, transportation, energy, and more.

Having capabilities at the edge means more than simply shifting resources to smaller devices networked together. It means discovering new paths to innovation and even new business concepts. Specifically, intelligent edge endpoints enhance the delivery of real-time insights, on-the-spot analytics, and services.

Enabling such capabilities at the edge requires tools and technology that allow businesses to quickly react to new opportunities as they emerge. Open source software is uniquely positioned to help developers build software that can seamlessly navigate heterogeneous hardware-software environments in an efficient and performant way, supported by an extensive ecosystem of silicon manufacturers, software vendors, and standards groups.

CHAPTER 2

Why Open Source Software Is the Best Fit for the Intelligent Edge

Open source software solutions offer several benefits for intelligent edge applications. To start, embedded systems for intelligent edge applications are typified by small footprints, high performance in a constrained environment, high reliability, and low resource requirements. All of these can be provided by a suitably selected enterprise Linux distribution.

Ecosystem

Open source solutions offer a wealth of community-developed coding, testing, and debugging solutions. Embedded systems used in intelligent edge applications need a Linux distribution with a compact core able to support low-resource environments, greater security requirements, and high performance. They also need the reliability to run independently in a remote environment.

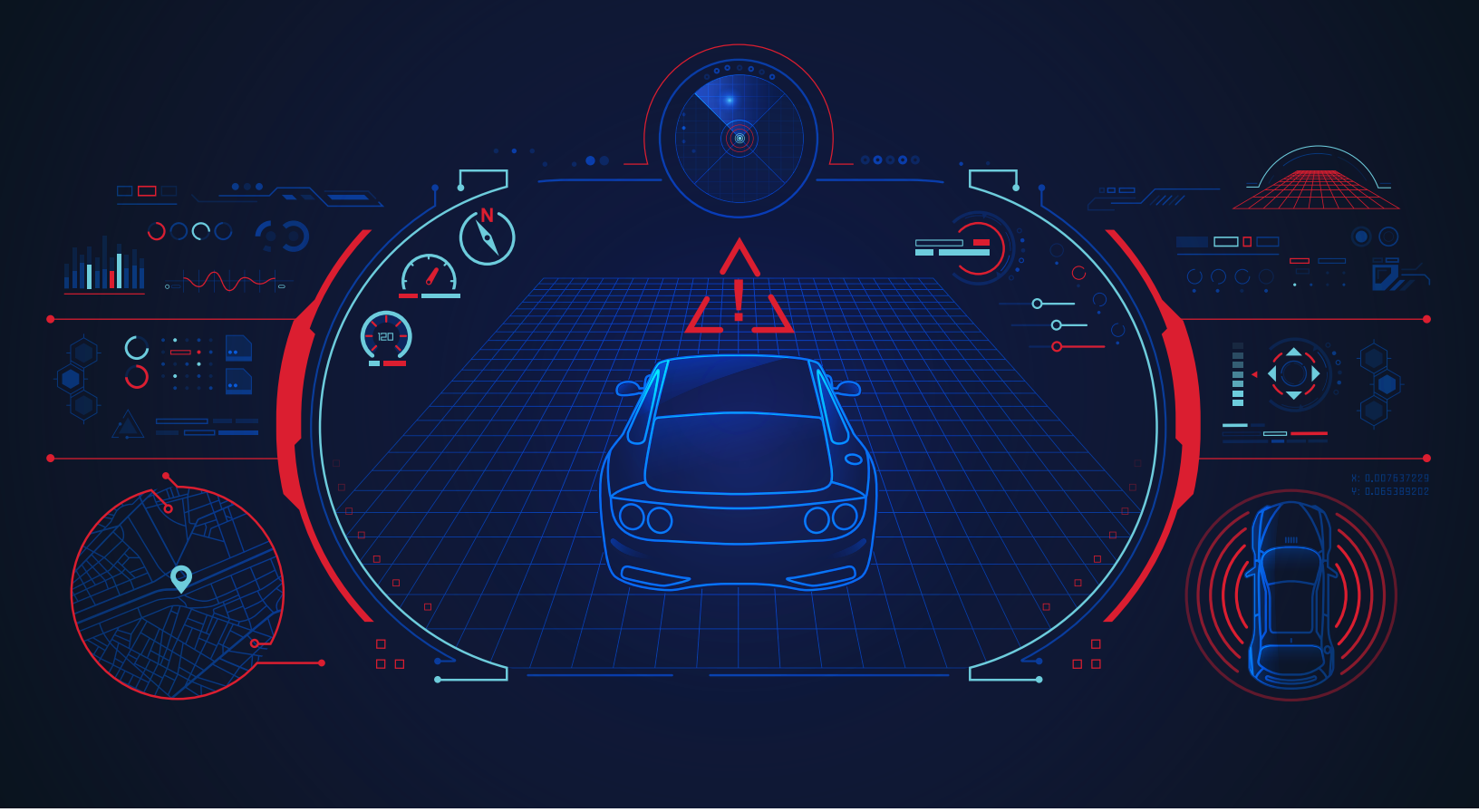
This core framework is generally provided by the OpenEmbedded and Yocto Project communities. It enables both economies of scale in terms of support and highly customized builds that exactly fit project requirements.

CI/CD

Many businesses and developers have embraced today's continuous integration and continuous delivery (CI/CD) development and deployment strategies. CI/CD bridges the gaps between development and operation activities and teams by enforcing automation in building, testing, and deploying applications. CI/CD forms the backbone of modern-day DevOps operations.

Open source supports CI/CD. The frequent software releases in open source communities allow DevOps to identify and resolve issues more quickly and with regular feedback from end users. A continuous delivery schedule is key to enabling the improvement of development processes with:

- Greater flexibility and stability
- Higher quality and efficiency
- Faster access to the latest features
- Tighter feedback loops



Innovating with AI

The intelligent edge is ripe for new applications. Increasingly, many applications are incorporating artificial intelligence (AI). There are many cases in which AI running on an intelligent edge device is used to make a decision (e.g., AI-based video image analysis in an autonomous vehicle to avoid roadway obstacles) or spot an anomaly and send an alert (e.g., AI-based analysis of a device's performance data).

The overwhelming majority of AI algorithms starts in open source. A great example of the work being done by the open source community to use AI in intelligent edge applications is TensorFlow. TensorFlow is an open source software library for high-performance numerical computation primarily used in machine learning. Its flexible architecture allows easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs) and a range of systems from single desktops to clusters of servers to mobile and edge devices.

CHAPTER 3

The Three Key Elements of New Edge Applications: Security, Reliability, and Real Time

The great enthusiasm for the intelligent edge must be tempered with some realities, however. Edge systems are prime candidates for attack. They often are hard to manage, update, and support since they are highly distributed. And many systems are based on proprietary technologies, making them difficult to incorporate into innovative applications or complicating their own accommodation of newly available processes and systems.

Increasingly, open source solutions are seen as a way to address these challenges and let businesses build new-generation applications. Open source solutions keep pace with changing requirements. Backed by a community of developers, new ideas emerge all the time, and fixes to problems can be quickly realized. There are open source efforts specific to the embedded solutions market that is at the heart of the intelligent edge.

It's worth keeping in mind that developers often need to add enterprise-class features to make the solutions useful for production environments. This problem is increasingly solved by working with a technology partner that offers enterprise versions of open source software.

With an intelligent edge, remote or decentralized nodes of a system carry out data handling that may have traditionally been dealt with at a central point in a system. Specifically, intelligent edge applications analyze data where it is generated and either send an alert to other systems or initiate an action. They are at the heart of many distributed applications in a wide range of industries. As such, they need to be secure, reliable, and able to work in real time. Here are some factors to consider regarding those key elements.

Built-in security

Sophisticated cyberattacks are proliferating globally. Edge applications, being distributed and highly connected, are ripe for attack. Developers need to be sure that software running on edge systems has not been compromised before being deployed—that the system boots and operates securely, both initially and after every update.

The problem is that security is difficult to scale. The more complex the software and computer system, the larger the test matrix. Not testing at full scale can put the production system at risk, and the unrelenting demand for faster deployments exacerbates this risk. Unfortunately, the solution has often been to forego complete test coverage and test only for the most critical use cases.

Developers of critical systems need tools that can help them stay a step ahead of increasingly sophisticated attackers. System simulation technology provides an efficient and effective means of researching, analyzing, and testing a wide variety of attack methods and security countermeasures in a flexible and scalable environment, and in ways that would simply not be feasible with physical systems. In a world that is ever more dependent on the safe and reliable performance of interconnected systems, simulation gives cyber professionals a way to gain the upper hand.

Reliability

Intelligent edge devices are being called on to perform critical roles. They often operate in isolated environments under harsh conditions (especially compared to devices that operate in a typical data center).

Devices must endure extreme temperatures, vibrations, and impacts. Mobile environments may incur periods when systems have limited or lost connectivity. As a result, an intelligent edge system must be designed and equipped with software to keep operating regardless of the environment or situation.

Edge and real-time operations

A real-time system gives you determinism in the sense that you can predict how the system will behave from an execution-flow point of view. And you can predict this accurately in terms of execution time.

Embedded systems that support real-time operations are being used in many critical application areas. For example, they are responsible for processes in manufacturing facilities and machines of critical importance, such as autonomous automobiles.

By using open source software, developers have access to lots of code and can see how the community has adopted similar technologies.

When you have use cases that are compute-intense (e.g., machine learning), an open source framework or OS can be another good choice, side by side with an RTOS to match the constraint and control aspects of an application.

CHAPTER 4

11 Cybersecurity Must-Haves to Make Open Source Solutions Work for the Intelligent Edge

Edge projects often need to incorporate new connectivity services such as 5G and increasingly powerful analytics algorithms. There is a need for flexibility to accommodate and integrate these new technologies into embedded edge systems. There also is a need for speed in today's highly dynamic marketplace.

Open source solutions help address both issues. They benefit from a community that rapidly adjusts to changes, addresses problems, and enables innovation. However, when edge applications are put into production, enterprise-class features and support are necessary.

Here are some of the top properties to look for in an open source solution for secure embedded systems:

1. **Data-at-rest protection.** Software, data, and configuration files are protected when stored in nonvolatile memory, typically using encryption. Encryption keys are stored in security hardware. Using certified or industry-standard encryption algorithms such as AES, RSA, ECC, or SHA will help protect data at rest and prevent an attacker from gaining access.

2. **Authenticated or secure boot.** Software (including firmware and configuration data) is authenticated or decrypted before use. Be sure to use secure boot technologies to ensure that hardware only kicks off the intended and authentic software and not an attacker's malicious code.

3. **Hardware resources partitioning.** Hardware computing resources (processors, cache, memory, devices, and networks) are segregated. An architecture in which components are isolated via strong, hardware-enforced boundaries enables defense in depth, especially if interfaces between separated components are tightly controlled.

4. **Software containerization and isolation.** Software applications should be well defined, self-contained, containerized, and isolated. Such an approach prevents an attacker from using a vulnerability exploited in one piece of code to gain unrestricted access to the entire system.

5. **Attack surface reduction.** The best approach to minimize an embedded system's attack surface is to use a trusted computing base, minimize dependencies and code base, and limit or narrowly define interfaces. A way to accomplish this is to remove code and interfaces that are not absolutely required. So instead of simply deploying a monolithic Linux distribution onto an embedded device, cut out the drivers, features, and code that are not needed.

6. **Least privilege and mandatory access controls (MAC).** Embedded systems should be created where they provide users and applications with only the minimal set of privileges and access using robust MAC. If an attacker successfully compromises a system subcomponent or gains root-level access, the combination of least privilege and MAC greatly constrains the freedom to maneuver and blocks the attacker's ability to modify, disable, or disrupt system services.

7. **Implicit distrust and secure communications.** Communication received by a system from external sources should be expressly denied until the remote source has been authenticated. And secure (i.e., encrypted) communication should be required even after the other party has been authenticated. Implementing mutual authentication and encryption ensures that a system is only communicating with trusted entities (and not an attacker) and that nobody else can eavesdrop on what is being communicated.

8. **Data input validation.** All data received from untrusted sources should be validated before being passed into software applications. Validating data before use helps to ensure that external inputs cannot unintentionally interrupt or maliciously exploit system functionality, leading to compromise of the system.

9. **Secure software development, build options, and OS configurations.** Software applications and OS kernels should be compiled and configured with all available security options enabled and enforced. Employing such defensive coding practices can significantly decrease the number of possible attacks that can compromise one or more parts of a system.

10. **Integrity monitoring and auditing.** A system must perform ongoing integrity monitoring and audit logging of security events. These measures will identify when a device is being attacked or whether it has been compromised. These warnings can be used to stop an attacker before it is too late. Specifically, auditing and monitoring confirm that a system has been attacked, help quantify the damage, and enable a quick recovery.

11. **Customization.** General-purpose Linux distributions are intended for well-resourced, multi-purpose, and often multiuser use cases. They are typically configured to support a wide range of devices with a one-size-fits-most mentality. An alternative is a commercially supported embedded Linux OS. The major advantages of this approach are like those of enterprise Linux, except that instead of creating a massive, one-size-fits-most distribution, most commercially supported embedded Linux vendors favor creating a distribution optimized for embedded systems.

If these properties are in place and implemented properly on systems, an edge solution based on open source software will have the resilience and security required for production environments.

Try Wind River Linux now by visiting labs.windriver.com.

RTInsights.com
Accelerate Your Business with Real-Time Insights

RTInsights is an independent, expert-driven web resource for senior business and IT enterprise professionals in vertical industries. We help our readers understand how they can transform their businesses to higher-value outcomes and new business models with AI, real-time analytics, and IoT. We provide clarity and direction amid the often confusing array of approaches and vendor solutions. We provide our partners with a unique combination of services and deep domain expertise to improve their product marketing, lead generation, and thought leadership activity.

WIND™

Wind River® is accelerating digital transformation of critical infrastructure by delivering the technology and expertise that enable the deployment of safe, secure, and reliable IoT systems.