WHITE PAPER

How Industrial Linux Enables Distributed IIoT Applications

Ryan Teng
Project Manager

MOXA®

Executive Summary

Distributed applications are on the cutting edge of the Industrial Internet of Things (IIoT). By leveraging the latest advances in edge computing, that is, deploying edge gateways to collect and preprocess data from numerous sensors and other devices spread across many different field locations before transmitting mission-critical information to the cloud, distributed IIoT applications bring the benefits of IoT to the remotest regions on Earth.

Besides satisfying industrial-grade hardware specifications, edge gateways used in distributed IIoT applications also demand a robust yet lightweight and highly customizable operating system for bespoke IIoT application development. This white paper discusses the role of edge computing in distributed IIoT applications, identifies the major challenges to implementing distributed IIoT applications, and explains how adopting an industrial Linux operating system can overcome these issues.

Released on November 29, 2019

Moxa is a leading provider of edge connectivity, industrial networking, and network infrastructure solutions for enabling connectivity for the Industrial Internet of Things (IIoT). With over 30 years of industry experience, Moxa has connected more than 57 million devices worldwide and has a distribution and service network that reaches customers in more than 70 countries. Moxa delivers lasting business value by empowering industry with reliable networks and sincere service. Information about Moxa's solutions is available at www.moxa.com.

How to contact Moxa
Tel: 1-714-528-6777
Fax: 1-714-528-6778

MOXA®
Reliable Networks ▲ Sincere Service

On the Edge of the Industrial Internet of Things

In recent years, growing investments in industrial markets have spurred rapid expansion in "Internet of Things" (IoT) application development. In fact, two-thirds of developers surveyed in a 2019 Eclipse Foundation study[1] revealed that their organizations already develop and deploy IoT solutions or plan to do so within the next 18 months. Even though IoT developers work in many different focus areas, industrial automation remains one of the top three[2] industries in 2019. More specifically, the global market for IoT gateways, which are placed between edge systems and the cloud, is expected to be worth as much as US$1.4 billion in 2021[3], and may even reach US$11.1 trillion by 2025[4].

The Industrial Internet of Things (IIoT) promises to revolutionize global manufacturing by leveraging data from interconnected smart sensors, industrial equipment, and analysis tools to improve production processes. Although many IIoT applications adopt a **centralized** architecture where communication devices connect to a central node (for example, smart factories that connect all the PLCs, actuators, and other industrial equipment to a central SCADA system through communication gateways and industrial Ethernet switches), **distributed** IIoT applications are on the rise[5].

In a distributed IIoT application, sensors and equipment deployed across a wide area connect to one of many edge gateways located throughout the entire network. Each edge gateway acts as a data concentrator, protocol converter, and data preprocessing device for all the sensors and equipment that connect to it. The edge gateway then transmits all of the preprocessed information from the edge system (comprised of the gateway and connected sensors and equipment) to a public or private cloud for big data analysis. Typical distributed IIoT applications include smart cities (such as smart meters and street lighting management), renewable energy (such as solar or wind farm monitoring), and oil and gas.

---

[1] Eclipse Foundation. IoT Developer Survey 2019 Results. https://iot.eclipse.org/resources/iot-developer-survey/iot-developer-survey-2019.pdf

[2] The top three industries and focus areas for IoT developers in 2019 are platform (34%), home automation (27%), and industrial automation (26%). Eclipse Foundation.

[3] Technavio (2017). https://www.businesswire.com/news/home/20170420005801/en/Global-Industrial-IoT-Gateway-Market---Drivers

[4] McKinsey Global Institute (2015). Unlocking the potential of the Internet of Things. https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/the-internet-of-things-the-value-of-digitizing-the-physical-world

[5] Jennath H.S., Adarsh S., Anoop V.S. (2019). Distributed IoT and Applications: A Survey. In: Krishna A., Srikantaiah K., Naveena C. (eds) Integrated Intelligent Computing, Communication and Security. Studies in Computational Intelligence, Vol. 771. Springer, Singapore.

The critical role served by edge gateways in distributed IIoT applications also illustrates the importance of **edge computing**, which essentially moves IoT data processing and actuation from the cloud to the edge of the network. By introducing a layer of gateways between the IoT devices (edge systems) and the cloud to preprocess the data, edge computing reduces latency for real-time applications, efficiently utilizes bandwidth and storage resources, enhances scalability, reduces costs and energy consumption, and improves privacy control[6].

Besides sufficient processing power and industrial-grade hardware requirements, edge gateways in distributed IIoT applications also need a robust and secure operation system. IoT developers particularly value operating systems that include common features and enable them to concentrate on business outcomes. In 2019, the most popular OS for IoT gateways was clearly Linux, with 76% of IoT developers using a Linux distribution for edge system development, compared to only 52% of IoT developers using a Windows platform[7]. To understand why Linux distributions are such a popular option for IoT edge gateways, let's examine the specific challenges faced by a typical distributed IIoT application and how the edge gateway OS can help.

Challenges of a Distributed IIoT Application

Distributed IIoT applications face unique challenges that need to be considered when choosing the development platform for the edge gateway. Consider the classic example of a digital oil field, which is usually located far from civilization and includes many oil wells scattered over thousands of acres[8] to pump underground petroleum to the surface. In order to provide predictive maintenance, real-time monitoring, alarm notifications, and other add-on value in a digital oil field application, all the information from numerous oil wells, pipelines, and other processing facilities needs to be collected and transferred to a public or private cloud server for big data analysis.

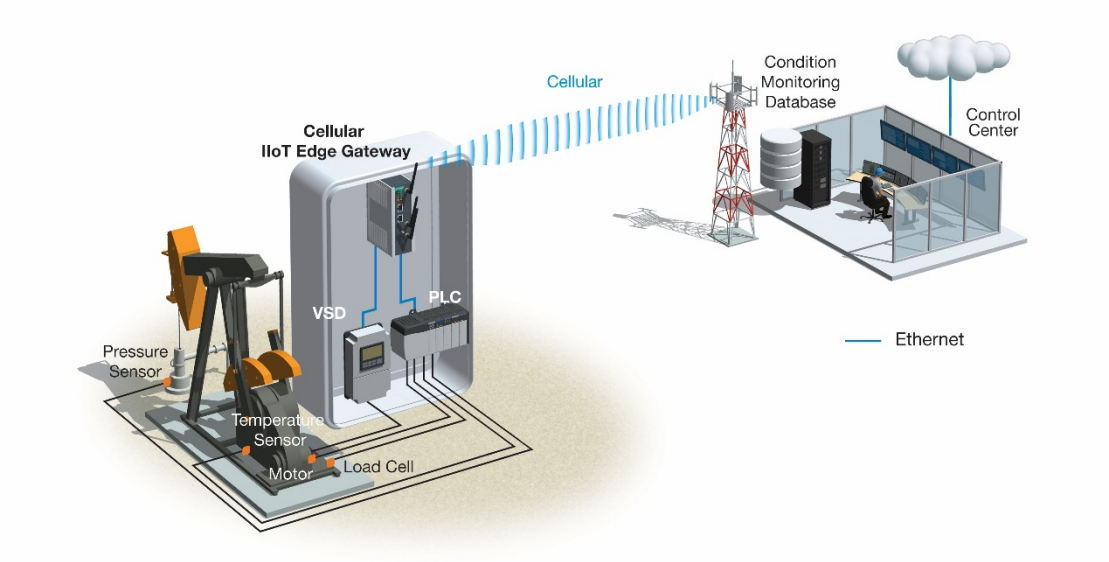
---

[6] *Canonical (2018). Ubuntu Core and Kura: A framework for IoT gateways.* *https://pages.ubuntu.com/rs/066-EOV-335/images/Kura_WhitePaper_29.10.18.pdf*

[7] *Eclipse Foundation. IoT Developer Survey 2019 Results.* *https://iot.eclipse.org/resources/iot-developer-survey/iot-developer-survey-2019.pdf*

[8] *The world's largest oil field, the Ghawar Field in Saudi Arabia, spans an area of 1.3 million acres (5,300 square kilometers). Sorkhabi, Rasoul. "The King of Giant Fields". GEO ExPro, Vol. 7 No. 4 (2010).* *https://www.geoexpro.com/articles/2010/04/the-king-of-giant-fields*

Due to the highly remote and distributed nature of oil field applications, wired communication is often difficult to deploy and maintain. Instead, digital oil fields often use cellular communications, or another wireless technology, by installing an edge gateway—along with I/O, PLC, and other devices—in an explosion-proof cabinet at each remote site. To ensure reliability in harsh oil field environments, the gateway should also have a wide operating temperature range and explosion-proof certification[9].



Besides the aforementioned mission-critical hardware requirements, edge gateways in digital oil fields and other distributed IIoT applications also require an operating system that addresses the following challenges.

Application Development

IIoT edge gateways need to perform several different functions and process large amounts of data from many different sensors and actuators at each oil well in real time. Since most gateways are only designed to process incoming data before transmitting the information to more powerful servers in a remote data center or the cloud, a real-time operating system (RTOS) is usually embedded on the microcontroller unit (MCU). However, the traditional embedded RTOS is usually designed for a single purpose, which makes it difficult to simultaneously perform multiple functions in real time. In addition, a simple RTOS is unable to support machine learning, containers, and other new technologies. Ultimately, these limitations delay time-to-market and increase overall development costs.

System Stability

Since edge gateways need to process mission-critical data at each remote site, the gateway operating system needs be incredibly stable because a system crash could easily endanger production or even human life. In general, there should be zero tolerance for system crashes or damages resulting from adding, modifying, or deleting files on the gateway. If an exception occurs or a software application freezes the system, maintenance engineers should be able to roll back the operating system to the last working version.

---

[9] *To learn more about specific explosion-proof certifications for the oil and gas industry, visit: https://www.moxa.com/en/solutions/industry-focus/oil-and-gas*

Remote Maintenance

All operating systems require periodic firmware updates and vulnerability patches. However, updating the firmware on so many gateways in such a remote and highly distributed application presents another challenge for digital oil fields. How is a maintenance engineer supposed to update the firmware on so many different devices in so many different locations? Physically travelling to each remote site is incredibly time-consuming and costly given the size of most oil fields. Moreover, how does a maintenance engineer ensure that the entire digital oil field system remains online and running if a single firmware update fails at one remote site?

Data Protection

All the oil well data stored on each gateway also need to be protected because the information is highly sensitive and confidential. Even if an intruder is able to steal the storage media, such as a flash drive or SD card, from the gateway, the data should be protected from unauthorized access through reverse engineering. Furthermore, the gateway operating system software should be protected and validated for the integrity. For example, if it is possible to bypass the normal boot-up process and replace the operating software, the gateway can be commandeered by unauthorized personnel and compromise the entire oil field system.

Future Support

Most operating systems are only supported by vendors for several years. However, unlike commercial applications in office environments that can upgrade to a new version of an operating system every couple of years, industrial applications like digital oil fields generally need to use the same platform for 10 years or possibly longer. After all, industrial applications run highly specialized programs for complex processes that require a great deal of time to implement and deploy. What's more, these programs may not even be fully compatible with new operating system versions.

Industrial Linux Distributions for Distributed Applications

Fortunately, new industrial Linux platforms can address the previously discussed challenges plaguing distributed IIoT applications by providing an open software platform specifically designed for industrial automation. However, as open-source initiatives supported by many different vendors and contributors including Moxa[10], industrial Linux distributions also vary substantially. As a result, it is important to choose an industrial Linux distribution that genuinely satisfies the following requirements for edge gateways in distributed IIoT applications.

[10] *To learn more about Moxa's contributions to the industrial Linux initiative, visit: https://www.moxa.com/en/spotlight/industrial-computing/arm-linux-iiot-edge-gateway-portal/linux*

Fast Time To Market

Due to the drastic drop in silicon costs in recent years and technological advancements enabling smaller chips to perform increasingly powerful computations, embedded operating systems for gateways are no longer limited to a single purpose or a simple RTOS. In fact, modern embedded systems are capable of running multipurpose Linux operating systems on edge devices, including IIoT gateways. The ability to support multipurpose functions enables IoT developers to focus on business outcomes and bring applications to the market even faster. In addition, choosing an industrial Linux operating system based on Debian, the most popular Linux distribution for IoT developers[11], can also speed up time-to-market by providing a familiar platform for developers to start adding on value.

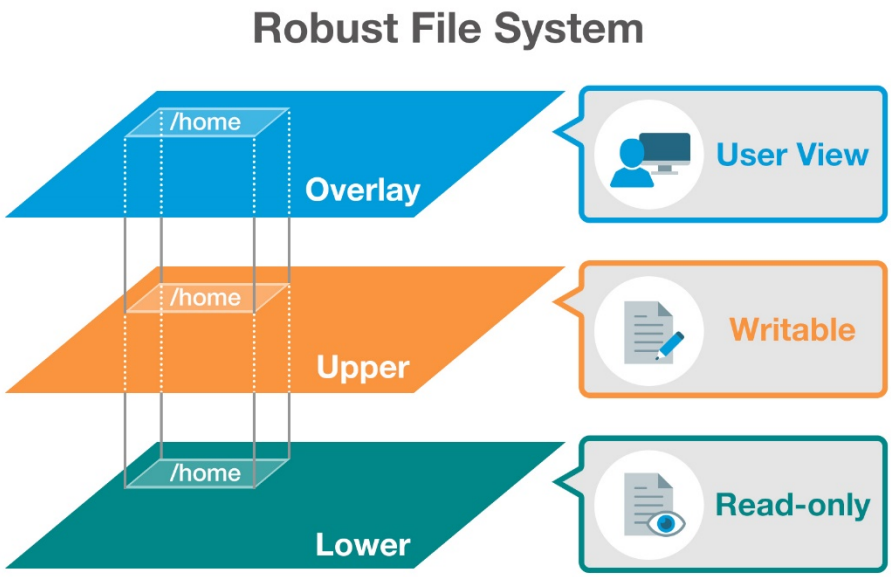
	Former Embedded OS	Modern Embedded OS
OS Type	RTOS/Embedded	General Linux
Toolchain	Cross-compiler	Normal toolchain
System Functionality	Single-purpose	Multi-purpose
Storage	Constrained SRAM, flash	256 MB RAM, 2 GB SD
Remote Updates	Few updates	Constant updates
Network Connection	Offline	Connected
Development Effort	Expensive and custom software development	Low-cost and simplified software development

Robust File System and Dual System Design

Industrial Linux platforms have three system layers: the bootloader, kernel, and file system. The most frequently changed layer during application development and operation is the file system. To prevent system crashes, the industrial Linux OS should provide a mechanism to prevent the file system from crashing and allow administrators to roll back the system to a previous version. More specifically, the robust file system should support the following:

- Firmware downgrades, in addition to upgrades
- Overlay File System (OverlayFS) to prevent system crashes caused by unexpected power loss during firmware upgrades/downgrades, or when restoring the system to default settings
- File system recovery if firmware upgrades/downgrades fail

[11] *Eclipse Foundation. IoT Developer Survey 2019 Results.* https://iot.eclipse.org/resources/iot-developer-survey/iot-developer-survey-2019.pdf



The industrial Linux OS should also incorporate a dual system design that retains the last working version of the bootloader or kernel if a bootloader/kernel upgrade fails. For example, administrators may need to upgrade the bootloader or kernel to patch a security issue or fix a bug. However, if the bootloader or kernel upgrade fails, the entire system will not be able to boot up, bringing the entire industrial system to a halt.

Over-the-air Software Updates

Because edge gateways are located at remote sites, it is difficult for administrators to upgrade the application and system software in the field. Remote firmware upgrades over cellular, Wi-Fi, or another type of wireless network provide the most practical way to overcome this issue. Debian systems in particular support a simple software upgrade mechanism called APT. APT, which stands for Advanced Package Tool, has a central repository of over 25,000 software packages ready for remote download and installation. Developers can even package their own security patches, bug fixes, or new application software in the APT format and provide the APT package to a central server, such as the device management server, to perform firmware upgrades over the air. To ensure that the APT package is genuine and comes from the original vendor, the device should also have a mechanism for validating and authorizing the APT server.

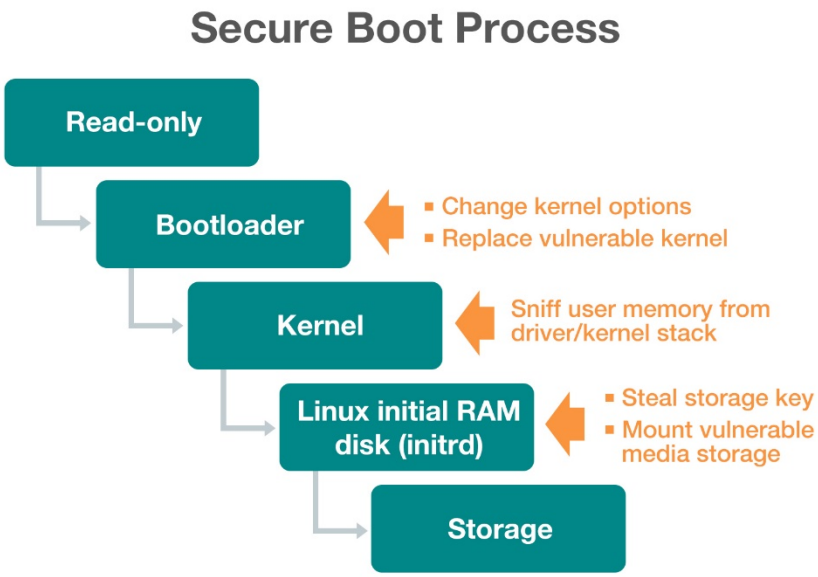
Built-in Industrial-grade Cybersecurity

The industrial Linux platform should have a built-in secure boot process to protect mission-critical data. By anchoring each boot process to the hardware root of trust (RoT), industrial Linux platforms can prevent the trusted computing base (TCB)—that is, the bootloader or kernel—from unauthorized access, thereby protecting the gateway from data theft or brute force attacks.

Secure boot requires a CPU that supports either IBM eFuse or Intel Boot Guard technology. Both of these technologies essentially hard-code critical programming logic onto a chip that cannot be modified after manufacturing. Generally speaking, the following OS boot processes are anchored to the hardware:

1. The CPU loads the bootloader[12]
2. The bootloader loads the kernel[13]
3. The kernel loads the mini root file system (initrd/intramfs)
4. The mini root file system mounts rootFS (ext4/raid disk)

In order to protect each thread from unauthorized code injections or sniffing, asymmetric cryptography and signature verification should also take place as each process is executed. The following figure shows a thread model for a typical secure boot process.



Besides storage protection during the OS boot process, secure boot should also include library protection for application software and binary data. For example, if an attacker physically steals the gateway from a remote site, he or she should still be unable to access mission-critical information because all the data and libraries have also been asymmetrically encrypted.

---

[12] Here, we use "bootloader" to refer to either the Arm "bootloader" or x86 BIOS interchangeably.

[13] On x86 systems, the bootloader runs the GRUB/LILO process before loading the kernel.

Long-term Linux Support

One of biggest concerns developers have about using a Linux distribution is the maintenance and support period. As with standard Linux operating systems, industrial Linux platforms are also open-source and may only be maintained by the original developers for about two years. For industrial applications, however, upgrading or migrating platforms after two years is unacceptable. Ideally, IIoT application developers should work with a software vendor that provides long-term support for their industrial Linux platform, such as the 10-year support offered by Moxa Industrial Linux, to extend the period of software maintenance and adjust the type and frequency of software updates (patches) to reduce risk, expense, and disruption to software deployment.

Conclusion

Distributed IIoT applications, such as digital oil fields, require robust and secure operating systems for edge gateways to preprocess mission-critical information before transmitting data to the cloud. Capable of addressing the application development, remote maintenance, data protection, and future support challenges affecting distributed IIoT environments, it is no wonder industrial Linux distributions have become the most popular edge gateway operating systems among IoT developers.

For example, Moxa Industrial Linux (MIL) is a small footprint, high-performance, industrial-grade Linux distribution that accelerates the development of embedded and IoT applications, and comes with 10-year Linux support that includes security patches and bug fixes, making industrial projects secure and sustainable. In addition, Moxa is a member of The Linux Foundation® and is part of its Civil Infrastructure Platform (CIP) project that aims to create an open-source platform for managing and monitoring smart cities, civil infrastructure, and factories.

Whether you choose MIL or another industrial Linux distribution, it is important that your edge gateway operating system enables fast time-to-market, includes robust file systems and dual system design, supports simple and secure updates over the air, has built-in industrial-grade cybersecurity, and provides long-term support.

For additional information on Moxa's industrial-grade Linux operating system for IIoT applications, visit: www.moxa.com/MIL