Why device management in the IoT matters and how to achieve it

By Keith Shea, Wind River

This article outlines the business case for efficient device management and introduces a solution for managing edge devices remotely, reliably, and cost-effectively.

For most enterprises, the compelling case for the Internet of Things (IoT) is the ability to access the valuable data being generated by hundreds or even thousands of field devices. That can happen only if the devices delivering that data and the gateways that direct data to enterprise systems are continually performing as expected. Device manufacturers and IoT system developers need to think upfront about how to manage those devices.

Data may be the hero of the IoT story, but the real workhorses are devices at the edge of the IoT system - the things in the Internet of Things. They're out in the field either generating and transmitting data to a centralized platform or performing automated tasks that generate data. A mundane job, perhaps, yet the overall performance of a system often hinges on the health of field devices. If a device, sensor, embedded agent, or gateway begins faltering, the consequences can be dire.

The challenge of maintaining devices may sound basic compared with aggregating and analyzing data, but it's essential to a successful IoT strategy. At a minimum, device manufacturers and system operators need a way to monitor the health of devices in the field to prevent system disruption and downtime. More importantly, they need to have an action plan: how to remedy those problems that will eventually occur. With IoT, change is constant. Business priorities will shift as companies gain insights about their operations from the data. So system operators need an efficient, scalable way to provide updates across a large fleet of devices. Security, too, is a major concern. If vulnerability is discovered in device software, patches must be deployed quickly - before intruders can exploit the gaps.

Device manufacturers and system developers need to plan for these contingencies at the design stage. With potentially thousands of field devices in play, it's not feasible or cost-effective to rely on truck rolls for fixes and updates. Instead, what's needed is a way to perform these tasks remotely, at scale, and over the Internet. But IoT data collection typically runs just one-way—from device to cloud. Even when operators detect device anomalies, they typically don't have the tools to push commands back to the device and fix the issue. So the initial design of an IoT system must consider the entire operating lifecycle, from deployment to decommissioning. Several distinct but interrelated issues must be addressed.

Once devices are deployed and connected, operators need a way to activate and provision them efficiently. Today, that often means physically going from device to device and loading applications or performing upgrades manually. IoT system operators need to be able to configure, provision, and manage field devices remotely. Device security is critical to an IoT system. Hackers often target endpoint devices as a means of gaining entry. And security breaches at the device level can have severe consequences: financial losses, damage to credibility, and even endangerment of human life. But securing devices is challenging since they're vulnerable to both physical tampering and network-borne threats. System operators need the right tools to monitor remote device performance and check for security vulnerabilities. They also need to be able to send instructions to those devices to correct a problem or change a function. This requires full two-way communication, where responses to devices can be completely automated.

Historically, information technology and operational technology systems have been kept separate. But IoT systems need to be integrated, with a centralized place to aggregate, analyze, and store data. While the devices in enterprise applications can perform for years, the software running on them will require regular updates and upgrades: from bug fixes to security patches to overall software improvements. And once an upgrade or a new application is ready, operators need to be able to deploy it quickly and cost-effectively to

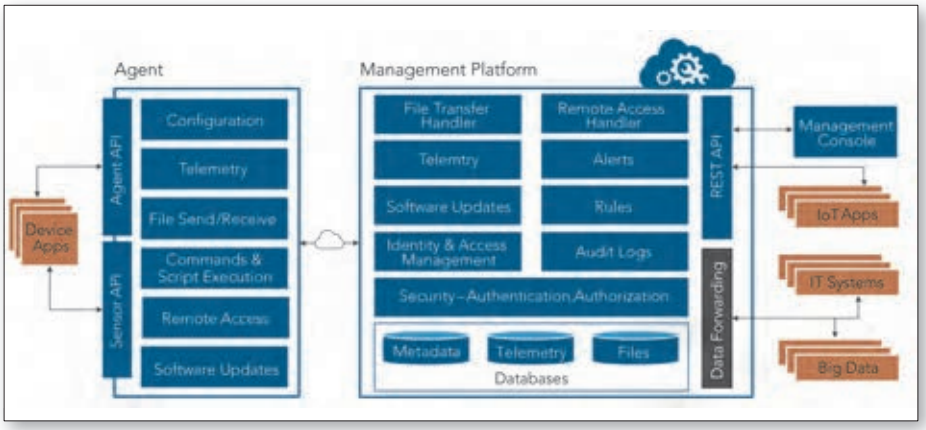

Figure 1. Block image of the Wind River Helix Device Cloud

many devices at once. Developers must plan for end-of-device life at the design stage so operators can easily and remotely remove a device from service.

The challenge facing every IoT system developer and operator is how to gain consistently reliable and secure remote control over devices typically far away and connected via the public Internet. Device management should be part of an IoT strategy from inception. But trying to build device management and two-way communication capabilities into a system from scratch can take time, devour resources, increase costs, and delay deployment.

A more practical solution is to leverage technology designed specifically for IoT device deployment and management. Wind River Helix Device Cloud is the ready-built platform that makes it possible. The solution also provides RESTful APIs, enabling IT and OT professionals to quickly build vertical-specific IoT solutions and integrate disparate enterprise IT systems. With Device Cloud, industrial companies can easily build device management capabilities into their infrastructures and greatly reduce the complexities of rolling out large-scale device deployments. Device Cloud gives customers the following abilities.

Deploy: connect devices to the cloud. Devices can be provisioned via a startup.bin file, authenticated via certificate exchange, and configured via network settings in the OS.

Monitor: record device-related information. Data is collected on device health (CPU, memory, etc), operations (pressure, speed, etc), connection status, and device alerts, for example.

Service: diagnose and repair devices remotely. Device application log files and historical trend data are analyzed, a tunnel is established to allow secure, remote device access, and repair procedures (change settings, push updates, etc) are conducted, when necessary.

Manage: track device properties and changes. The agent reports device properties and other "inventory" information that may be useful for understanding what is running in the field.

Update: deliver content and software updates. Updates can be made to files, application software, the agent, and even the OS kernel.

Decommission: remove devices from the system. Devices are stopped but agent files remain (deactivate), device is returned to factory default state, or devices may be deleted from the cloud, and all device data is erased (decommission).

Device Cloud automatically collects and integrates data from disparate devices, machines, and systems, enabling operators to track device status, share data, and proactively determine when updates are needed. Using an embedded software agent, device properties and operating data can be transmitted securely to the cloud. Operators can easily view device information through a web-based management console, perform diagnostics, and take prompt corrective action. The cloud-based platform is also designed to integrate with enterprise systems that utilize or analyze data from IoT networks. Device Cloud data and event forwarding ensures that device health issues will signal other systems of potential problems, allowing them to respond accordingly and prevent ingestion of potentially bad data.

Recent security breaches with connected field devices have brought the urgent imperative to protect connected systems to the forefront of the IoT conversation. Security is imperative for IoT applications, for the protection of the machines they control, and for the people who depend on their reliable performance. Further, an industrial company's success hinges on securing their connected devices and their data. Effective security requires an end-to-end strategy that spans the entire application lifecycle.

Security adds a additional layer of complexity. Without proper planning, building in security functionality can slow down development, drive up costs and, in some cases, impair the performance of a deployed application. With Device Cloud, users can build IoT applications on a platform using pre-configured, integrated software components in which many security issues have already been addressed. This takes the onus off developers to identify, source, and patch together different security technologies as development progresses, resulting in a much more efficient development process, much less system complexity, and a reduced risk of security gaps due to misconfiguration. Device Cloud includes a wide range of pre-configured features that enable developers to implement security measures across the device lifecycle at the design stage, including: secure boot, device software update mechanism, SPM, application whitelisting, network, data, and device encryption, embedded credentials and certificates, Trusted Platform Modules, access permission, software isolation, and integrity measurement. By providing pre-integrated security components, Device Cloud helps developers mitigate the risk of misconfiguration and implement security without delaying development or compromising system performance.

With IoT adoption becoming widespread, a growing number of enterprises are unlocking the valuable data generated by their everyday operations: gaining business insights, optimizing operations, improving profitability, and uncovering new business opportunities. But IoT can only be effective if connected devices are actively monitored and managed. Fortunately, technology exists that makes it easier to build that capability into IoT devices and systems. Utilizing Device Cloud, device manufacturers and IoT system developers can accelerate device deployment and close a critical gap in IoT operations, ensuring that the devices enterprises depend on for crucial business data are secure, responsive, and performing at the highest possible level. ■