Managing your IoT devices

A guide to selecting IoT device management protocols and features that solve your IoT device management challenges

Anna Gerber

January 03, 2018
(First published December 22, 2017)

In this article, discover how the IoT device management protocols and features help you address many IoT device management challenges, including scalability and availability among others.

IoT 301: Mastering IoT development

This article is part of the IoT 301 learning path, an advanced developer guide for IoT.

- Get serious
- IoT security challenges
- IoT device management (this article)
- IoT analytics
- Tutorial: Extend an IoT system

Any complex IoT system must include device management capabilities in its architecture. IoT devices are often deployed in hostile environments. And, when devices fail, they might need to be retired and removed from those environments or updated so that they can continue to operate in those environments.

Device management helps to protect devices and their data by making it easier to secure and monitor the devices. Device management capabilities allow IoT developers to control IoT devices by performing operations like resetting them to factory defaults or applying updates to patch security issues or fix bugs.

As the number and variety IoT devices deployed increases, and the complexity of your IoT system architecture also increases, managing your IoT devices becomes increasingly challenging.

Device management challenges

The key challenges involved with managing and maintaining IoT devices include security, interoperability, constrained devices, scalability, and availability.

Security

Security is an important consideration across all layers of your IoT system. Device management services can make it easier to secure the IoT devices themselves by providing secure device registration and authentication services and by supporting encrypted machine-to-machine (M2M) communication. Read more in my Top 10 IoT Security Challenges blog post on the developerWorks blog and how implementing device management services in your IoT system can address many of these security challenges.

Interoperability

The devices deployed within an IoT system now and into the future might be of different classes, be produced by a range of manufacturers, and use a range of communication protocols. Device management tools must support managing all of the devices consistently to maintain interoperability across heterogeneous devices. Look for device management services that support standard device management protocols, or which implement protocols and APIs to provide abstractions for managing devices generically in bulk.

Constrained devices

IoT devices are often constrained, which means they have limited power, memory, processing capability, or connectivity. (Read more about IoT devices in my IoT hardware guide.) These constraints affect whether the device is capable of being managed remotely, and how effectively it can apply remote operations. If a device is powered by a battery, it is vital that the device be able to communicate with the device manager and perform updates or operations without exhausting the power available. If the power supply is exhausted and then an operation such as a factory reset or firmware update is interrupted, it could result in bricking the device.

Lightweight device management protocols are ideal for IoT devices, as they are designed to be efficient and minimize the amount of processing that needs to be performed by the device itself. These lightweight device management protocols reduce the bandwidth and frequency of communication between the device and the management services so as to conserve device resources.

Scalability

As more devices are added to the system, device management services need to scale to handle larger numbers of devices that are registering and communicating with the device management service. Device management services need to be able to handle the increased number of routine device management operations that will need to be performed at any given time.

Automation is key to scaling device management for IoT, because it quickly becomes impractical to monitor and manage all of the devices manually. An administrator should only need to step in to intervene if something unexpected happens.

Availability

Device management services must be aware of context in order to ensure availability. Device management includes monitoring the current state of devices so that you don't try to reboot a device while it is in the middle of an update, but also includes broader awareness of the state of

the network, awareness of the device's status and available power, or what the current device usage looks like before performing maintenance operations. Device management services should support synchronizing management operations like firmware updates to minimize disruption.

Device management protocols

Many of these device management challenges can be addressed by adopting standard device management protocols or by making use of device management services provided by an IoT platform. (Read more about why you might need to use an IoT platform in your IoT solutions in my previous developerWorks article.)

IoT devices typically perform machine-to-machine (M2M) communication over lightweight connectivity protocols like XMPP (an XML-based chat protocol), CoAP (Constrained Application Protocol), or MQTT (MQ Telemetry Transport). Read more about IoT communication protocols in my connectivity and network protocols guide.

Device management protocols operate over the top of these general connectivity protocols, to support device registration, authentication, querying device capabilities, and performing operations consistently across devices.

Standardized device management protocols that have been applied to IoT devices from the broadband and mobile industries include TR-069, OMA DM and LWM2M:

- **TR-069.** The BroadBand Forum's TR-069 Customer Premises Equipment (CPE) WAN Management Protocol (CWMP) was originally developed in 2004, based on SOAP, for managing broadband equipment including modems, routers, gateways, and home devices including set-top boxes. This protocol has been applied within IoT smart home applications.
- **OMA DM.** The Open Mobile Alliance Device Management (OMA DM) specification was the predecessor of LWM2M, developed for mobile phones, PDAs, and tablets, and was first released in 2003. It is designed for constrained devices that have limited bandwidth, and supports M2M communication over a range of protocols including HTTP, WAP, or SMS. It can be applied to IoT devices to support provisioning, configuration, firmware updates, and fault management. However, it is not as lightweight as LWM2M.
- **LWM2M.** OMA's Lightweight Machine to Machine (LWM2M) protocol is designed to manage devices communicating over cellular networks, for example, within a sensor network. LWM2M is usually implemented over CoAP. There are a number of open source implementations of clients and servers that support the LWM2M protocol, including ARM mbed and Eclipse Leshan and Wakaama.

IoT device management is an area of active standardization, so this area remains quite fragmented. Many IoT reference architectures describe device management features, and most IoT platforms, such as the IBM Watson IoT Platform, implement custom device management services that have been tailored to the requirements of managing IoT devices for use with the specific platform.

Read how to connect a Raspberry Pi as a managed device in this developerWorks recipe.

The IBM Watson IoT Platform Device Management protocol is a lightweight device management protocol that operates over MQTT. The IBM Watson IoT platform supports both managed and unmanaged devices. Managed devices run a device management agent, which includes the logic for connecting to and communicating with the Watson IoT Platform's device management service.

Device management features

IoT platforms typically provide APIs and dashboards for their device management. These dashboards and APIs can be used to manage device registration, trigger remote operations, and to monitor, search for, or filter devices (for example, by manufacturer or serial number).

Whether you are considering adopting device management provided by an IoT platform, a standalone service (for example an Eclipse Leshan server), that implements a standard device management protocol, or a combination of device management services, key features to look for include:

- **Provisioning.** When a new device is added to the system, the device should securely register itself with the device management service, as well as registering the metadata for the device. Registering the device provides it with identity and credentials.
- **Authentication.** Authentication services establish the identity of devices. The device uses the identity created initially during the provisioning process, so that whenever it communicates with other devices, apps or services, the other party can be assured that the device is a trusted, authentic device.
- **Configuration.** Device management services typically support applying new configurations to IoT devices directly or by broadcasting new configurations to update devices in bulk, as well as managing device configuration dependencies.
- **Monitoring & diagnostics.** Device management services are also responsible for keeping track of device logs and metadata, for example, a service might track device capabilities, firmware version, the usual location of the device, device ID and status. Device management services often expose this information, along with error and connectivity logs through dashboards or APIs to use for monitoring health and status as well as for diagnostics and remote debugging. The logs and status can also be used to generate alerts, for example, if the device has not produced any data for a certain period of time.
- **Scheduling remote operations.** Many device management services support scheduling remote operations, including device reboot, enabling or disabling the device, performing a factory reset and triggering a new firmware download and update through over the air updates. Being able to perform these maintenance operations remotely without manual intervention, helps to save time and money throughout the life of the device, as well as helping to avoid mistakes and minimize device downtime throughout the process. It also removes the need to manually retrieve or update devices installed in locations that are difficult to access physically.
- **Automation.** Automation becomes a necessity when the number and range of devices deployed within an IoT system starts to scale. Automation helps to simplify applying remote operations in bulk, for example to rapidly perform a firmware update on multiple devices to address a security vulnerability. Depending on the device, these updates might be applied through specialized device management protocols, however for Linux-based devices,

Open Source orchestration tools like Kubernetes can be used to deploy Docker containers containing firmware, applications or operating environments across multiple devices.

- **Retirement.** Devices eventually fail, or are superseded when they come to the end of their service life. Device management services should support decommissioning devices securely, including revoking any tokens and identities associated with the device so that it is no longer able to communicate with other devices, apps or services within the system.

Conclusion

Device management services help to automate the management of IoT devices throughout their lifecycle - including provisioning, authentication, configuration, maintenance operations, monitoring, and eventually decommissioning. Device management is a critical component for any scalable, secure and interoperable IoT solution.

Related topics

- Device Management Protocol docs
- developerWorks recipe: Device Management in Watson IoT Platform
- IoT 101: IoT hardware guide
- IoT 101: IoT platform guide
- IoT 101: Networking technologies guide
- IoT 201: IoT reference architectures
- Series: Design and build secure IoT solutions