IoT essentials - enabling IoT Device Management

1. INTRODUCTION

It was around 30 years ago when John Romkey turned the first toaster on over the Internet with TCP/IP & SNMP protocols. Now there are more “things” connected to the Internet than people living in the world.

Nowadays, connecting a toaster to the Internet is not as difficult as it was 30 years ago. However, connecting thousands of small sensors with flash memory so low they can barely run an operating system, and at the same time securing the communication between them and the server - now that can still be a challenge. This whitepaper will explain the fundamental part of any IoT ecosystem – **IoT device management**. You will find out how you can easily and securely manage the device lifecycle with **OMA LwM2M** (OMA Lightweight Machine To Machine) as well as how using the leading device management standard can help you create a successful IoT deployment.

2. TODAY'S IOT ENVIRONMENTS

Due to the rapid growth of IoT deployments in various industries and the ambiguous use of terms such as 'IoT platform' or 'IoT solution', it can be a challenge to determine which exact aspects are crucial in order to create a successful IoT ecosystem.

Among the obvious fundamentals is providing the necessary hardware (devices, sensors etc.) and connectivity to build the solution. Following this, it is essential to provide **the ability to manage** (e.g. upgrade the firmware), **control and monitor these devices.** We also can't forget about the need to analyze the collected data and provide a secure communication channel between the devices and the platforms. On top of that, the presence of scalable data storage, aggregation and orchestration capabilities as well as some kind of visualization features so that all the collected data can be presented in a user-friendly form (dashboards with charts etc.) make up the usual elements of IoT deployment. Of course, to ensure greater flexibility in integration of third-party systems, elements such as application programming interfaces (APIs), software development kits (SDKs) and gateways are very useful.

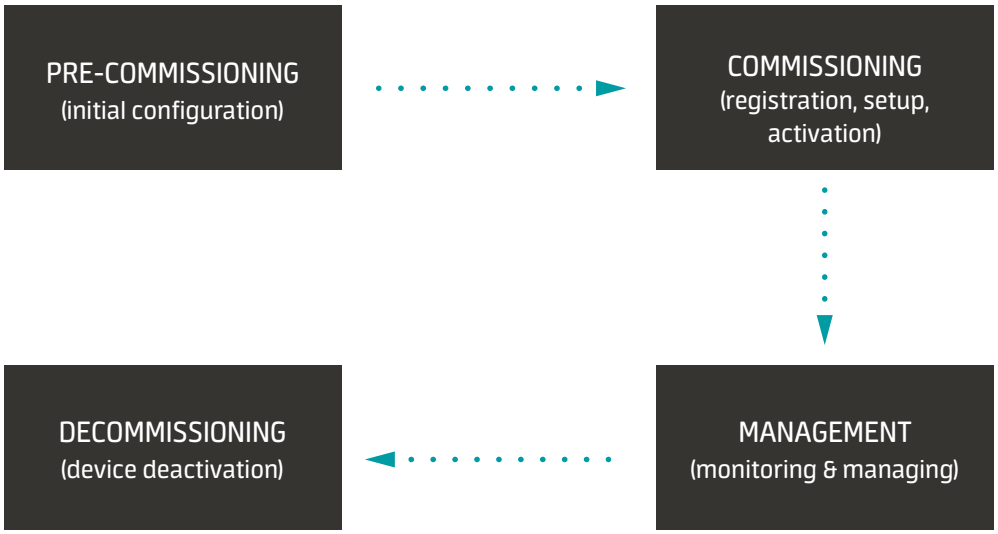
A well-thought out IoT ecosystem may also include features such as business intelligence or analytics allowing it to analyze data in real-time and thus, for example, predict the need to perform maintenance of devices. Most importantly, however, **in every IoT deployment it is crucial to watch over all the connected devices**, check if there are problems with firmware/software updates and initiate actions for real-time responses. All of which is possible thanks to device management. Only then, when proper management of devices is ensured, can the other IoT software components even be considered.

3. WHAT IS IOT DEVICE MANAGEMENT?

The Internet of Things is expanding at full throttle and controlling the ever-increasing number of connected devices slowly becomes its greatest challenge. The connected devices, home appliances, vehicles, sensors, actuators and whatnot need to be monitored and updated with new firmware or software, reconfigured, sometimes even decommissioned etc.

Modern IoT device management should be capable of **taking care of devices through their entire lifecycle** – from provisioning devices so that they can register with the server and stay connected, to allowing for easy removal of the device from the service when its life has come to an end.

Some of the most important key features covered by IoT device management are pre-commissioning, commisioning, management, decommisioning.



Pre-commissioning

The IoT device is usually pre-provisioned before being deployed and connected. The pre-provisioned information typically includes a specific set of credentials that will in the future allow a device to connect with a potential IoT platform. It is also a security measure because it enables the device to be authenticated by the system or registered with the server. After the authentication is done, the system can identify the device on the network and block all the other 'unauthorized', potentially dangerous devices.

Commissioning

The next phase is commissioning which takes provisioning of the device one step further. Thanks to the pre-commissioning phase, the device can now be authenticated. Once the authentication is done, the server identifies the device and the registration of the device becomes possible. When the device is registered with the server, we finally gain control over it and can start configuring it, establishing telemetry link for data collection, provisioning it with additional information specific to our needs and remotely managing the device. It is worth mentioning that in the case of resource constrained devices **the commissioning phase can be reduced to FOTA** after which the device switches to sleep mode for a specified period of time.

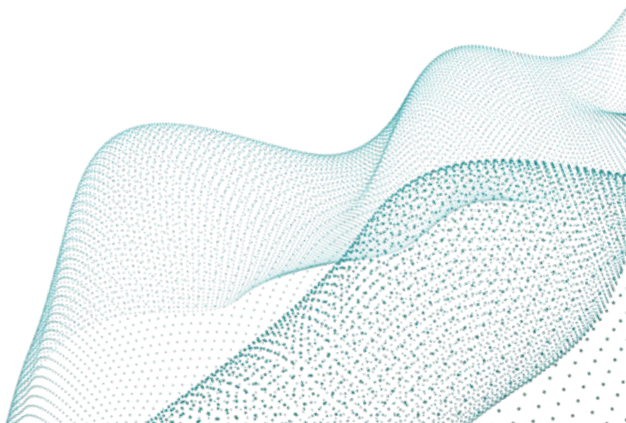
Management

When the device has been finally registered and the communication between the system operator and the device has been established, it is time to prepare for ongoing device management.

Devices can be very powerful and can require a complicated environment to work efficiently. Or they can be resource and energy constrained and wind up in a hard to reach place. At any rate, **it is very important that the devices are regularly managed and preventively monitored.** Over-the-air firmware updates and upgrades of software (FOTA/SOTA), remote monitoring & diagnostics, configuration updates, and ensuring security are among the most necessary functionalities provided by device management systems.

Decommissioning

Finally the last (but not least) feature that every device management needs to include is secure decommissioning of a device. Such decommissioning should involve deleting all the data (especially sensitive and confidential data including security certificates and any other credentials or shared secrets). Selective decommissioning from the whole fleet of devices is not an easy task but with proper device management even this process can be automated and performed remotely.



4. THE IMPORTANT ASPECTS OF DEVICE MANAGEMENT

Scalability and flexibility

It is important to plan ahead and prepare for device management on **a large scale**. This way we can avoid the need to restructure the whole system and keep the management efficiency if our deployment grows to an enormous size. It is also worth considering the lifespans of the devices and the longevity of the technologies chosen.

To ensure flexibility it is important to create clear distinctive groupings of devices and an easy and comprehensive way of collecting and aggregating all the data so that the device management tasks such as firmware updates can be easily performed and automated. The crucial factor here is the **ability to define automated actions with your own set of conditions** in regard to all the data received (be it from a single device or from the whole group).

Security

Another important aspect of device management in the IoT that has already been mentioned before is security.

Despite the obvious fact that all the devices creating IoT solution should be secured, security in the IoT tends to be neglected. Security problems, similar to the problems of scaling up when the number of connected devices rapidly grows and gets out of hand, seem to be symptomatic to the newly started businesses. As has been seen over the years in various more or less dramatic in consequences examples, flaws such as unencrypted communication and lack of a thorough authentication process can result in devices being easily compromised by hackers. Such cases can lead to a loss of control over the devices and obviously, serious problems for the end-users.

Problems like that only become more severe as time passes and should be overcome rather than overlooked. **Security should be taken seriously right from the start even in the smallest of IoT deployments.**

Integration

As the IoT gains popularity, more and more protocols, technologies, platforms, services etc. emerge on the market. As a result it gets even more difficult to address the diverse integration requirements of IoT deployments. Such integration involves not only data integration or back-end system integration but usually also third-party middleware integration.

A good idea would be to let APIs take care of scalable integration with a little help of built-in integration capabilities offered by your IoT platform. If every asset were to be a controlled API, the visibility of the flow of data would be greatly increased making your devices easily manageable and secure. At the same time, there is always the need of being prepared for a use case which cannot be handled by APIs and that needs to be addressed through a commercial integration solution. The important factor here is to correctly assess if you are going to need an additional solution to integrate all the components of your IoT deployment. This of course requires knowing exactly which data needs to be collected and analyzed to provide valuable outcomes for each particular use case.

Interoperability

Many devices in the IoT are managed via proprietary protocols because proprietary software is believed to have more features that appeal to the consumer (although usually it is only because of the large market share that seemingly inspires reliability and professionalism) than standards-based software. From the consumer's perspective you always want to end up with the most compatible and efficient solution for your appliances. Thus, if you have purchased a device from vendor X you will most probably have it working over the solution X. The problem begins when a consumer wants a smart TV set that works with solution X and a smart coffee maker that works with solution Y to communicate with each other.

Most importantly however, deployments in the enterprise world are the best example of interoperability as an essential element of the IoT ecosystem. **Deployments of carrier-grade services are in great need of unified and interoperable device management due to the different verticals coming into play as well as the large number of different vendors of devices using different technologies.**

Unfortunately, there is no universal standard in the IoT that would enable full interoperability between devices and we are by no means close to achieving this kind of utopia where all devices understand each other and are able to efficiently intercommunicate. However, there are standardization bodies that develop useful and interoperable Internet standards. One such organisation is **oneM2M** which aims to create a common M2M service layer by integrating various interoperable solutions. An area that has been addressed very well with an interoperable standard (also covered within oneM2M) is IoT device management thanks to **OMA SpecWorks' LwM2M protocol** becoming the established standard for managing devices in the Internet of Things.

5. DEVICE MANAGEMENT STANDARDS

Some standards are too heavy for most IoT applications. Some handle lossy nodes and networks in more efficient ways than others. Eventually some are optimized specifically for device management. Understanding which protocol to use for which application is not an easy task. Such choice usually involves selecting the best bandwidth requirements, real-time performance or memory footprint of a standard. Fortunately, there are a few truly efficient standards in IoT device management that make the choice of the protocol easier due to their multilayered structure and versatility. Some of them include layers that offer great help in device management and provide a well defined data model which significantly helps to ensure interoperability between a vast array of different devices.

LwM2M

One such standard that was already mentioned is **Lightweight M2M (LwM2M),** the standard developed by **Open Mobile Alliance (OMA) SpecWorks** with its version 1.0 published in 2017. LwM2M was designed especially for resource constrained devices (although basically any device can be managed via LwM2M). The standard is also quite effective over unstable connections and low bandwidth networks such as sensor or cellular networks.

The communication between the client and the server is based on User Datagram Protocol (UDP) working over Constrained Application Protocol (CoAP) with the support of data formats such as Type-Length-Value (TLV) which enable fast parsing and small data size. LwM2M utilizes Datagram Transport Layer Security (DTLS) as its security protocol to ensure authentication, confidentiality and data integrity. The big role in security mechanisms in LwM2M plays the bootstrapping interface which offers bootstrap modes responsible for provisioning

devices with essential information that would later on enable the client (device) to perform registration with the server and start the communication flow between them.

The pioneers of device management

As far as pioneers of device management are concerned, the most popular ones were OMA Device Management (OMA DM) with its first version published in 2003 and TR-069 also known as CPE WAN Management Protocol (CWMP) published in its first form in 2004. These two protocols, however, are mostly applicable in telecommunication services solutions.

MQTT

In the IoT, a standard that is often mentioned alongside LwM2M is the Message Queuing Telemetry Transport (MQTT).

MQTT has also been created with resource constrained devices in mind. It is a publish/subscribe messaging protocol suited for low-bandwidth and unreliable networks. Unlike LwM2M, it has continuous session awareness due to the use of Transmission Control Protocol (TCP) as a delivery protocol. While TCP is generally considered to be a more reliable way of communication, in some IoT deployments, especially ones that involve resource constrained devices, UDP is valued more due to its lightweightedness and faster transmission of packets.

What is worth mentioning is that the data model in MQTT is undefined and is usually specific to project or vendor requirements. Due to such flexibility, for example in the choice of payload, MQTT can be quite lightweight. It has to be noted, however, that LwM2M is still going to have smaller payload thanks to data types like TLV and use of User Datagram Protocol (UDP) as a transport layer.

Despite being a very light and compact protocol with link consumption smaller than MQTT, LwM2M offers much stronger security mechanisms determined by very strict requirements. Although in MQTT it is recommended to use encryption and authentication via Transport Layer Security (TLS), this has a considerable impact on the link consumption and client's performance.

Finally, it is important to note that **MQTT is an M2M connectivity protocol rather than a device management standard**. Due to its undefined semantics, MQTT, similarly to proprietary solutions such as Amazon Web Services (AWS), has mechanisms that ensure successful implementations but **lead to proprietary lock-in** situation where customers are dependent on vendor-specific products and services.

	LwM2M	MQTT
Transport	UDP, SMS	TCP
Application Layer	CoAP	-
Payload	TVL, JSON	Undefined
Link Consumption	Low	Depends on payload encoding
Data model	Defined	Undefined
IPv6	Yes	Yes
Security	DTLS 1.2	SSL/TSL
Market presence	Newcomer	Wide adoption
Standarization body	OMA, IETF	OASIS, ISO

6. CONCLUSION

In order for the IoT deployment to be successful you need to **plan ahead with device interoperability** in mind. If you want your devices to last for many years you should consider managing **the whole device lifecycle** including pre-commissioning, commissioning and decommissioning of devices.

This is where device management comes in response to these needs. The correct choice of the device management solution can not only solve the problem of scalability and flexibility but also provide your devices with high-class security and protect your investment over time. The fact that there have been numerous device management solutions and standards emerging throughout the years reflects the importance of this element as a must-have component in any IoT ecosystem.

About Coiote IoT

Coiote IoT is a portfolio of IoT solutions provided by AVSystem that answers the needs of device management and data orchestration in Internet of Things environments.

COIOTE IOT_
Device Management

Coiote IoT Device Management Platform offers complex device management features and telemetry data handling using the LwM2M device management protocol. The product allows for automatic detection of new devices and enables, among others: remote configuration, monitoring, user management, FOTA/SOTA upgrades, provisioning devices (with the option to group them to enable easy mass task execution etc.).

Apart from flexible workflow mechanisms, user-friendly GUI and REST API, Coiote IoT Device Management offers the ability to be integrated with other platforms and systems, such as the Coiote IoT Data Orchestration platform.

COIOTE IOT_
Data Orchestration

Coiote IoT Data Orchestration is an integration platform that enables the collection and orchestration of data received from various sources (devices and platforms). The end goal is to use existing infrastructure to create new supplementary services for various IoT verticals with simple and automated workflows. The platform can be used to provide advanced workflow mechanisms, monitoring and alerting, Business Intelligence, map visualization and customizable dashboards. The solutions that are based on Coiote IoT Data Orchestration enable connectivity management, asset tracking and service activation.

ANJAY LwM2M SDK_

AVSystem's Coiote IoT portfolio is complemented by Anjay LwM2M SDK, an open-source solution for enabling Lightweight M2M on any kind of device.

About AVSystem

AVSystem is a provider of software solutions for various industries. The company is especially known for their expertise in the area of large-scale IoT device management, multi-protocol device management for the telecommunications industry as well as WiFi Value Added Services. AVSystem is also an evangelist of open-standards such as LwM2M and TR-069 and provides embedded client solutions for hardware manufacturers. 100+ customers prove the superiority of AVSystem's solutions.