Overcoming security challenges and gaining interoperability with LwM2M device management

1. INTRODUCTION

As recent Gartner reports state, it is said that more than 20 billion IoT devices will be online by 2020. This rapid growth has raised concerns about many aspects of implementation methods associated with connecting devices to the network.

The challenges IoT environments are facing include scalability, security and interoperability, among others.

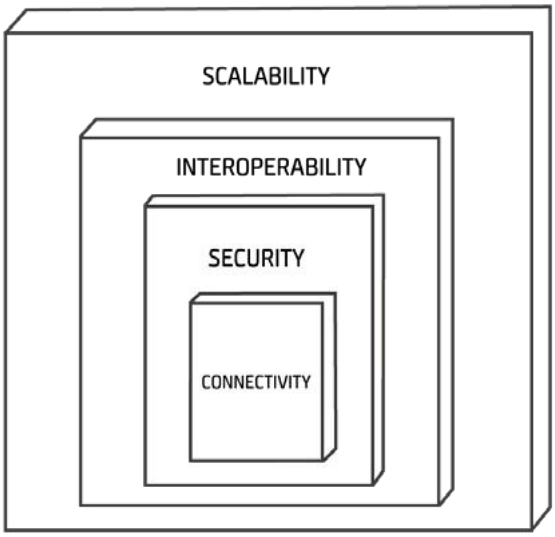


Figure 1. Challenges in the Internet of Things

Due to the fact that connected devices take part in almost every aspect of our lives, improving their security should definitely be of great concern to any IoT service provider. However, it is important to note that **interoperability is closely connected to security** because the data in the IoT industry is sent through various channels including sharing between many systems as well as sending it into the cloud and back. It is thus crucial that such data is always secured. The critical factor of communication between devices is not only exchanging the data but understanding the context of the data - the whole semantics and syntax behind it.

2. SECURITY – A KEY CONCERN

Taking into account the number of devices and the growth of the market, a potential breach in security may result in a problem of global scale.

One example of the IoT security breach is the infamous Mirai botnet attack. Mirai was a malware that turned Linux based devices into remotely controlled bots. It was used in a large scale botnet attack on numerous connected devices such as home routers or IP cameras. The Mirai virus infects internet devices by using a table of more than 60 common factory default usernames and passwords to find those that are still using their factory default username and password and log into them. Infected devices will continue to function normally, except for occasional sluggishness, and an increased use of bandwidth. A device remains infected until it is rebooted. After a reboot, the device will be reinfected within minutes unless the login password is changed immediately. The reason why Mirai was so dangerous is due to its ability to infect countless insecure devices and coordinate them to mount a massive DDOS attack to take advantage of a chosen victim.

Main security issues

Another important point is that while exposure of data generated by a smart coffee maker does not seem very harmful, when combined with data generated by other smart house appliances, it can provide quite sensitive information about the user's life pattern in an alarmingly intrusive manner.

However, not only privacy raises concern. There are a lot of questions about legal issues associated with IoT devices which include for example data flow across borders. The other important fact is that the customer needs to have control over their personal data being collected, stored, used etc. There have also been discussions about anonymity or secondary use of collected data, which indicates that there is a clear need to upgrade the quality of users' consent standards. **The client has to not only know how to choose and control the flow of their own data but also how to withdraw consent whatsoever.**

It is thus very important to address such problems right at the start, and it should be of interest not only to network operators but also businesses, enterprises or the government.

One of the reasons why security issues related to devices arise could be the **location of the device**. Some of the devices end up in hard to reach environments. Such unmanned devices which would have to be remotely updated could suffer from firmware being compromised. On the other hand, the devices stationed outdoors might corrode vecause of the weather conditions or simply get stolen. The other important factor that should be considered is the longevity of the devices. Some of them have life spans of more than 10 or even 20 years (while others will not last longer than 2 years). Long device lifetime can be problematic if the technology becomes obsolete or the device changes its owner, not to mention the fact that the older the device, the more prone to damage it becomes.

Insecure IoT endpoints are also a result of **scalability issues**. The problem is that not all IoT devices are created with security in mind from the beginning. Or they might simply lack storage capacity or computational power that is necessary to apply these solutions.

3. INTEROPERABILITY AND SECURING IOT DEVICES

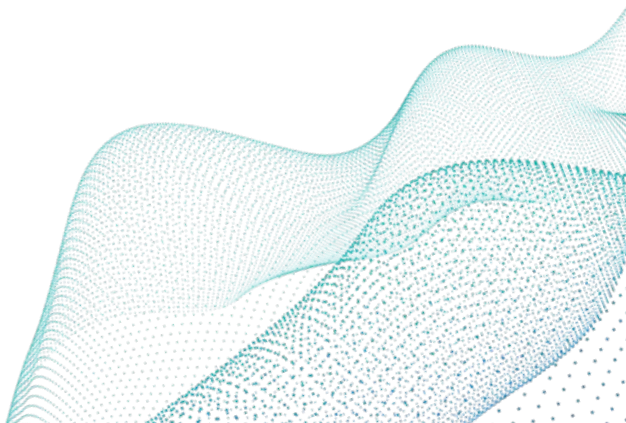
The fact that concerns about interoperability overlap with security threats mentioned earlier proves that interoperability remains one of the biggest challenges the IoT is facing.

The significance of the problem is best reflected in the number of IoT platforms and proprietary frameworks that have been emerging in the course of time. Utilizing different technologies and solutions results in limited integration possibilities between devices. The increase in number of connected devices also causes the fragmentation of IoT into more categories, which makes the interoperability even more difficult to achieve.

Therefore **dealing with security threats by focusing on interoperability is crucial to solving problems such as the legacy challenges or inconsistency in standards.**

However, it is important to understand that interoperability does not mean having one single standard that would unify all the solutions and approaches we have today (which would be ideal but in reality remains highly unlikely) but rather building common interfaces, semantics and solutions that would be easy to integrate with each other and flexible enough to cover any kind of use case. Such an abstract meta-layer that would allow to generate any kind of solution that is needed should be developed as a **layered standard** so that one could pick up an adequate protocol for each layer from the available stack to suit one's particular use case.

One protocol that offers much freedom when designing an IoT project is MQTT. It does not enforce any type of data thanks to its application-specific payload which results in overall simplicity of usage. Unfortunately, MQTT does not offer interoperability on higher layers like the application layer. The data model is undefined and the semantics layer can be designed differently for every single deployment (thus, not helping out with the issue of interoperability). There are, however, organizations that are working on solutions which treat interoperability as the most important aspect, one of them being **OMA SpecWorks and their LwM2M protocol for device management.**



4. STANDARDS-BASED APPROACH

OMA SpecWorks has developed a protocol especially designed for device management in the IoT industry. **Lightweight M2M (LwM2M)** is a standard that specifies device management and service enablement mechanisms designed especially for resource constrained devices (although basically any device can be managed via LwM2M). The technology functions very well over potentially unstable and low bandwidth networks such as cellular or sensor networks.

In spite of the fact that LwM2M comes with a few already defined objects and resources, it is not enough to provide structured data models which would allow efficient communication. Responding to these needs is the IPSO Smart Objects project (now also part of OMA SpecWorks) that aims to ensure the interoperability on the application layer by creating and providing a common design pattern of objects based on LwM2M. I**PSO Smart Objects** are the great extension of the already existing object model proposed by LwM2M Enabler because they were designed in particular to be reused and easily adjusted to new cases.

Furthermore, there is also a possibility of defining your own objects with a special editor provided by OMA and register them with **OMNA (Open Mobile Naming Authority)**. The objects will then be reviewed by individual members of OMA and added to the registry (if they pass the evaluation).

5. LwM2M SECURE DEVICE MANAGEMENT AND TELEMETRY

Even though the standard is particularly dedicated to resource constrained devices, security features are one of its key benefits. It is based on Constrained Application Protocol (CoAP) and utilizes Datagram Transport Layer Security (DTLS) as its main security mechanism. Along with UDP and SMS transport channel bindings, DTLS implements authentication, confidentiality and data integrity between the Server and the Client.

There are 3 security modes: certificates, raw public keys, pre-shared keys.

The Client needs to have credentials and the configuration information obtained during the bootstrapping process. Once the bootstrap is done, the server authenticates the Client, and thus the Client can make use of all the security features supported by LwM2M.

Figure 2 depicts a simplified presentation of the bootstrapping process and the registration of the client afterwards.

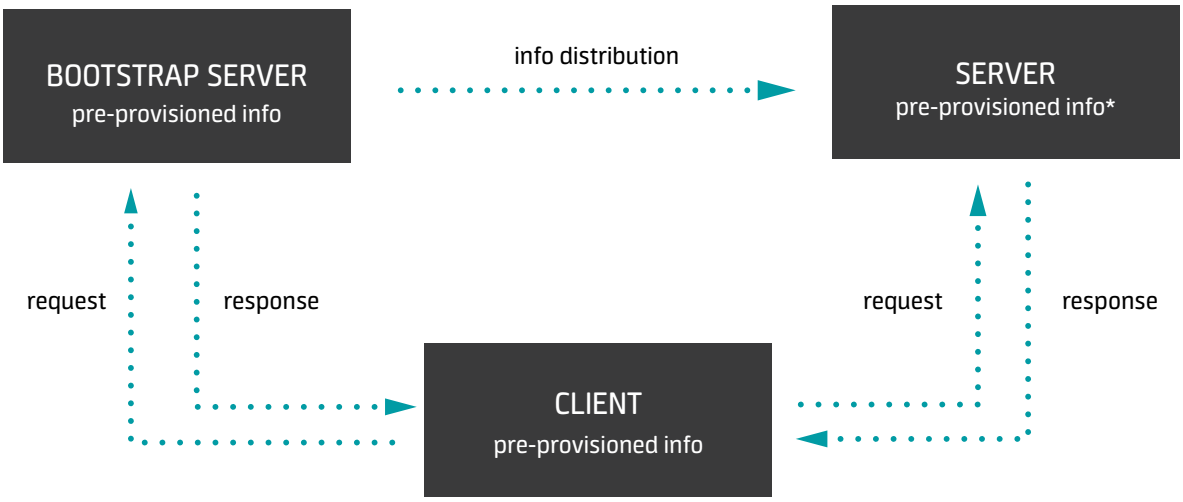
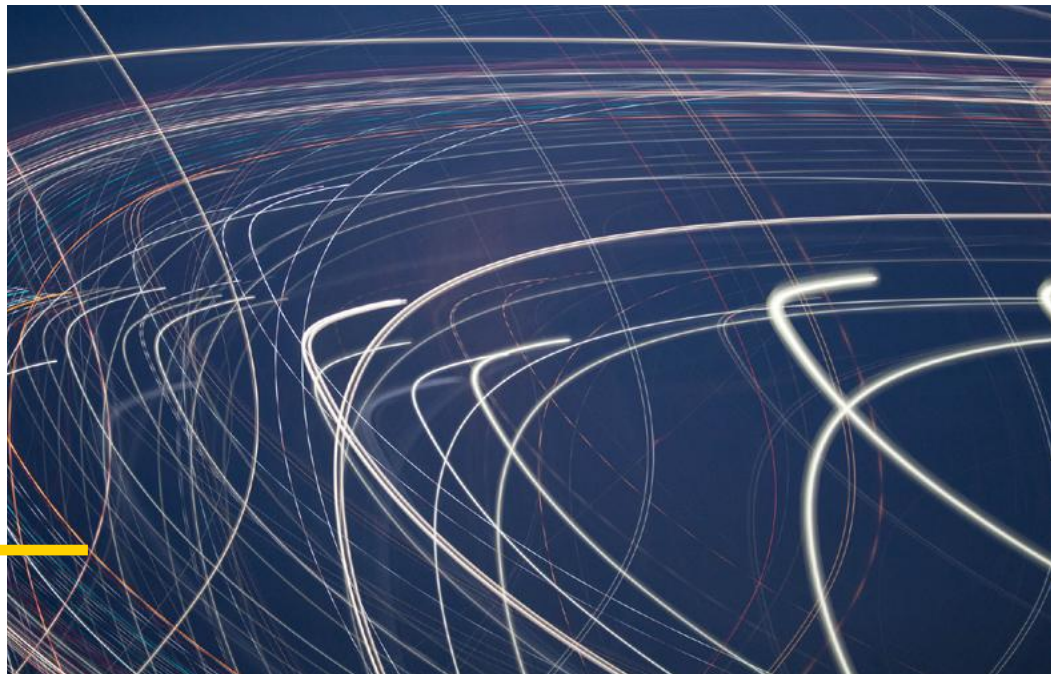


Figure 2. LwM2M bootstrapping process and client registration

The LwM2M Client pre-provisioned with information needed to initiate bootstrapping process sends the Bootstrapping Request (along with the needed credentials) to the LwM2M Bootstrap Server. The LwM2M Bootstrap Server receives the information sent by the LwM2M Client and sends the Bootstrapping Response (containing the information needed to initiate the registration process) to the Client while at the same time providing the LwM2M Server with the information needed to authenticate the LwM2M Client's registration request. After the bootstrapping process is done, the LwM2M Client and LwM2M Server can effectively communicate with each other due to the information shared by the LwM2M Bootstrap Server.

The information sent between the LwM2M Client and the LwM2M Bootstrap Server, as well as the information sent by LwM2M Bootstrap Server to LwM2M Server, and the information sent between the LwM2M Client and the LwM2M Server varies depending on the security mode used.



Bootstrap modes

It is worth noting that LwM2M provides the user with a standard data model for configuring security such as setting keys or certificates. It is very helpful if one wants to quickly react to potential security breaches or change certificates regularly as a preventive measure. The strength of LwM2M's security are its bootstrap modes. There are **4 modes of bootstrapping supported by LwM2M: factory bootstrap, bootstrap from smartcard, client initiated bootstrap, and server initiated bootstrap.** The device with factory bootstrap comes pre-provisioned with credentials which should be enough to ensure security.

However, if there was a need to change any parameters, one would have to have physical access to the device which is usually very inconvenient in the IoT. This is where device/server initiated bootstrap is of service, because you can invalidate previous credentials any time you want and the device/server can trigger re-bootstrapping. Or you can re-configure a device so that it connects to a different server. The LwM2M Bootstrap Server can also generate and overwrite its own certificate which can be useful if there is a need to update any security parameters without having to schedule the whole firmware upgrade.

Alternative security measures

At the same time LwM2M also provides mechanisms for smooth and **secure FOTA/SOTA** process regardless of the actual location of a device. Usually transfers of bigger amounts of data happen due to firmware update of the Client. This kind of updates poses a few challenges especially for limited wireless network bandwidth. Sometimes the problem occurs when many devices download updates simultaneously and the network becomes overloaded. There are also cases where the transmission breaks during the downloading process or it takes too long for an update to be downloaded because of the slow connection.

Although by default firmware update is done via PUSH request which means it will be performed via CoAP by default. Thus, it is worth considering the possibility of conducting the update **via PULL request**. This allows us to make use of other protocols than CoAP. Being able to use for example HTTP(S) to send bigger files can considerably improve the efficiency of the whole device-server communication. The PULL request also makes it possible to provide firmware update from different hosts which means there could be one device specifically dedicated to provisioning of firmware update so as not to involve LwM2M Servers in the whole process. The ability to use more efficient protocols grants significant flexibility as far as server infrastructure is considered which might be useful for further optimization to get the most out of LwM2M technology.

Access control

Another important aspect of security in LwM2M is **access control.** It is used to determine what operations are authorized for which server. Supposing there were two LwM2M servers: one big platform managed by a telecommunication operator and the other installed in a smart home central unit. Bootstrap server assigns full access control to the telecommunication operator's server and at the same time limits the access control for the smart home central unit server to read-only, so as not to allow a potentially inexperienced user to accidentally break the whole installation. What is more, the technology also ensures a secured and controlled telemetry channel which supports defining simple logic such as data flow with specified conditions for every parameter.

5. CONCLUSION

The disruptive and rapid growth of the Internet of Things comes with many new opportunities and technologies. The multitude of protocols used and lack of interoperability between them remains one of the main problems in the industry. However, throughout the years this issue has been addressed quite effectively and now we have **open-source software**, universally supported languages, common communication protocols and many other solutions that show promise for improved interoperability.

The lifetime of endpoints and services in the IoT industry is not infinite and within that lifetime the security requirements and standards are bound to change. Interoperability provided by standards such as **LwM2M** is one of the steps to securing the IoT. Well managed and secure endpoints are one of the basic components that ensure the overall security of each IoT solution. Ultimately, as long as new technologies, protocols, products and services are constantly reviewed, the challenges related to interoperability and security will be overcome and the Internet of Things will be a success.

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.