No Harm to Networks

How oneM2M Standard-Based IoT Solutions Protect Mobile Networks

By
Bob Flynn, Chordant and
Miguel Rodriguez, Deutsche Telekom

As Internet of Things (IoT) Devices begin to grow in number and start leveraging new mobile network capabilities, Mobile Network Operators (MNOs) will need to plan how to protect their networks from potential harmful effects. There are a number of past scenarios where IoT devices have caused significant degradation in service quality, impacting not only other IoT devices, but affecting entire regional cellular networks, ultimately causing breaks in service availability.

The GSM Association (GSMA), an industry trade body that represents the interests of mobile network operators worldwide, has created guidelines for efficient IoT device connectivity [1]. Similarly, oneM2M™* open standard defines services and capabilities to ease the implementation and deployment of IoT devices and applications. In this article, we will summarize the GSMA guidelines and provide an overview of how oneM2M standard-based solutions can fulfill those GSMA requirements and achieve 'No Harm to Networks'.

I. Introduction

Typical IoT deployment architectures include an IoT server that offers a set of services to customer-facing IoT applications. Figure 1 illustrates an example of cellular IoT deployment. IoT applications are based on the data and information exchanged with associated IoT devices. Massively deployed, battery powered IoT devices do not generate large or frequent messages. For example, smart electric meters used in residential areas would only send their daily usage reports to the electrical utility service provider. However, there may be thousands of such smart meters deployed in a small geographical area of a given mobile network.

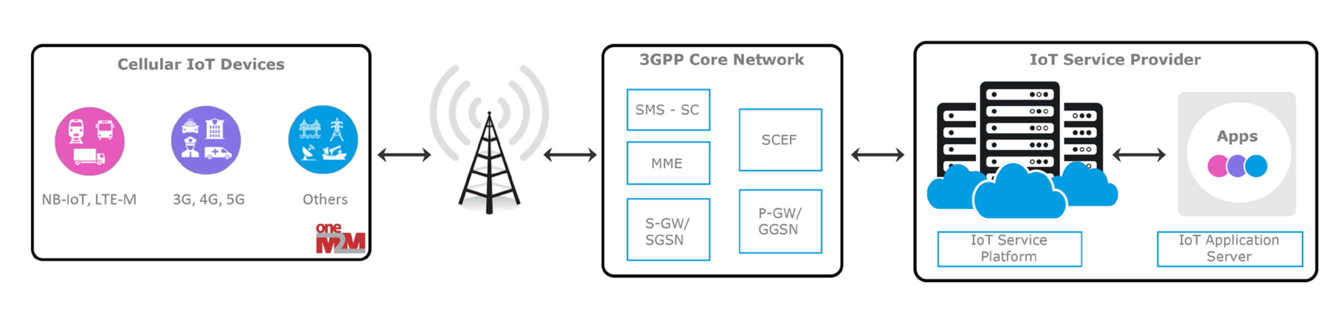


Figure 1- Cellular IoT Deployment

Unfortunately, IoT devices and IoT servers do not always operate as intended. Some examples of unexpected operations from actual deployments and their services include [1]: 1) a large numbers of devices attempting to access the network at the same time after a network outage; 2) aggressive attempts to obtain Packet Data Protocol (PDP) contexts; or 3) applications that use unintelligent error handling procedures. These unexpected operations can result in "denial of service" attacks. Other issues can be caused by insecure IoT application level communication, the use of fake or incorrect International Mobile Equipment Identities (IMEI), and algorithms that can lead to cascading attempts of devices rebooting and requesting network access. In general, most application developers do not have practical experience of cellular 3GPP procedures needed to account for these exception scenarios in their designs.

A recent example of such widescale potential harm to the network occurred in an enterprise IoT deployment that spanned 6 different countries [1]. It took the affected Mobile Network Operator approximately 48 hours to completely resolve this issue. The deployment consisted of 375,000 IoT devices that normally communicated to an IoT server via fixed line Ethernet connections. They only used the mobile network as a backup. The

*oneM2M is a trademark of the Partners Type 1 of oneM2M problem began when the IoT server became unresponsive. All the devices then attempted to use the backup mobile network. Since the IoT server remained unresponsive, the devices also reset their GSM communication modules, thus causing all the devices to try to re-register and establish new PDP contexts to communicate with the IoT server. This reboot loop created signaling overload in one of the home network's Home Locations Registers (HLR), effectively blocking all devices associated with that HLR from registering on the network.

Such cascading events underscore the necessity for the industry to create a set of requirements that all IoT service deployments should implement to prevent or recover from communication problems over a mobile network. These requirements are described in GSMA's ***TS.34 IoT Device Connection Efficiency Guidelines*** [1]. These guideline suggest that much of the risk posed by IoT applications can only be properly dealt with if one implements protection mechanisms on the devices themselves. Functionalities outlined in TS.34, such as Network Friendly Mode and Radio Policy Manager, proactively block harmful communication patterns from IoT devices. However, these solutions only mitigate damage to cellular networks without directly solving the root issue of poorly designed applications or compromised devices.

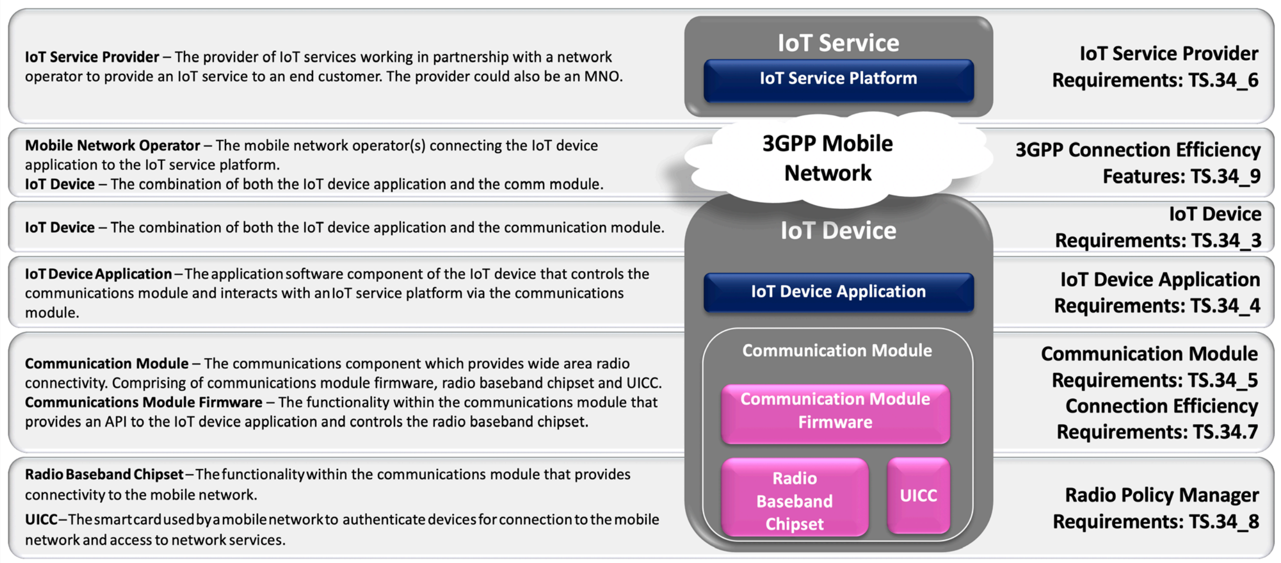
This is where the IoT service layer and service enablement functions can play a role, analogous to how operating systems on smartphones control what mobile applications may or may not do. The oneM2M standard defines such framework for an IoT service layer. oneM2M defines a standard-based approach that can be deployed in various hardware and software. The oneM2M standard offers common service enablement functions simplifying both device and application designs, such as a comprehensive security architecture and communications management. Release 3 of the standard began adding specific services for interworking a oneM2M service layer to the APIs exposed by mobile core networks, e.g. 3GPP Service Capability Exposure Function (SCEF) API's [4].

The remainder of this article is structured as follows. Section II provides an overview of the GSMA guidelines for efficient cellular IoT connections. Section III reviews difficulties that arise when applying these guidelines to massive scale IoT deployments. Section IV presents an overview of the GSMA proposed IoT architecture, evolved using the oneM2M standard principles. Section V describes specific oneM2M Common Service Functions (CSFs) that implement guidelines defined by the GSMA. Section VI outlines a deployment scenario that allows MNOs to manage devices on their network in a way that ensures efficient network operations. Section VII discusses how MNOs can certify IoT devices for use on their networks. Finally, Section VIII concludes the article by summarizing the benefits of using oneM2M standard-based solutions for cellular IoT deployments.

II. GSMA Guidelines for Efficient Cellular IoT Connections

In response to the numerous reported occurrences of cellular IoT deployments causing harm to the cellular networks, GSMA published ***TS.34 IoT Device Connection Efficiency Guidelines*** which define an optimal cellular IoT service architecture. This includes an IoT service hosted in the cloud and an IoT device application hosted on the UE (User Equipment or IoT device). Figure 2 shows the requirements for various components of such an IoT service architecture. These requirements fall into three broad categories: 1) congestion control handling procedures, where the device handles unexpected events in a manner that does not impact the operation of the mobile network, 2) communication management, where the device ensures that it operates according to communication policies which may be dynamic based on mobile network conditions, and 3) management of the device components and policies that control the device behavior.

It is important that IoT services, IoT service providers and mobile network operators implement these requirements to ensure that IoT devices do not harm the mobile network. While the focus of these requirements is to ensure that the mobile networks are protected, these requirements also benefit IoT service providers by improving the communication efficiency and performance of IoT devices and applications.



IoT Service Provider – The provider of IoT services working in partnership with a network operator to provide an IoT service to an end customer. The provider could also be an MNO.

Mobile Network Operator – The mobile network operator(s) connecting the IoT device application to the IoT service platform.
IoT Device – The combination of both the IoT device application and the comm module.

IoT Device – The combination of both the IoT device application and the communication module.

IoT Device Application – The application software component of the IoT device that controls the communications module and interacts with an IoT service platform via the communications module.

Communication Module – The communications component which provides wide area radio connectivity. Comprising of communications module firmware, radio baseband chipset and UICC.
Communications Module Firmware – The functionality within the communications module that provides an API to the IoT device application and controls the radio baseband chipset.

Radio Baseband Chipset – The functionality within the communications module that provides connectivity to the mobile network.
UICC – The smart card used by a mobile network to authenticate devices for connection to the mobile network and access to network services.

IoT Service
IoT Service Platform
3GPP Mobile Network
IoT Device
IoT Device Application
Communication Module
Communication Module Firmware
Radio Baseband Chipset
UICC
IoT Service Provider Requirements: TS.34_6
3GPP Connection Efficiency Features: TS.34_9
IoT Device Requirements: TS.34_3
IoT Device Application Requirements: TS.34_4
Communication Module Requirements: TS.34_5
Connection Efficiency Requirements: TS.34.7
Radio Policy Manager Requirements: TS.34_8

Figure 2 - GSMA Defined IoT Service Architecture

III. Challenges With Massive Scale IoT Deployments

MNOs can certify that communication modules operate “efficiently” on their mobile networks through some standardized verification procedures. However, problems start to emerge when IoT devices run various uncertified IoT applications communicating over the network using those cellular modems. Each of these unique applications can implement distinct, non-standardized communication behaviors or logic, that are “overlaid” on the cellular modem that the MNO had verified previously. Because of the seemingly endless array of potential scenarios, MNOs have an exponentially-growing challenge for IoT solution verification. It is unrealistic to assume that all IoT device manufacturers will follow the best design practices such as those recommended by GSMA since verifying compliance on various networks is both time intensive and expensive.

Another risk stems from the fact that ***TS.34 IoT Device Connection Efficiency Guidelines*** protection mechanisms for the Radio Policy Manager are not implemented by all suppliers in the same way. The feature is not configurable in many cases since it requires the MNO to implement custom files on their SIM cards. Thousands of read/write operations to the SIM card over time can effectively damage it. Also, some suppliers implement custom commands to enable or disable the features. However, this poses a security risk that the application can simply switch off the network protection feature altogether. This problem is unique to the IoT space, as smartphones and consumer products typically rely on operating systems from Google or Apple to control application functionality. Such applications also go through a quality-control process before being published for general usage. If we apply these concepts to the IoT space, verified application or service layer implementations could be paired with verified or certified communication modules, thus resulting in more robust and scalable IoT ecosystems. We look at how the oneM2M standard addresses this in the next section.

IV. Evolved IoT Service Architecture using the oneM2M Standard

GSMA recognized and described a future architecture for IoT services that reduces some of the challenges described above. An evolved IoT device architecture was proposed where a component called the "IoT Embedded Service Layer" provides generic IoT functionalities such as device management, security, location, and others. The architecture defined by the oneM2M standard is very much aligned with this evolved architecture view described by the GSMA. For example, the oneM2M standard defines an Application Service Node Common Service Entity (ASN-CSE) which is an IoT device software analogous to the GSMA "IoT Embedded Service Layer". This is illustrated in Figure 3.

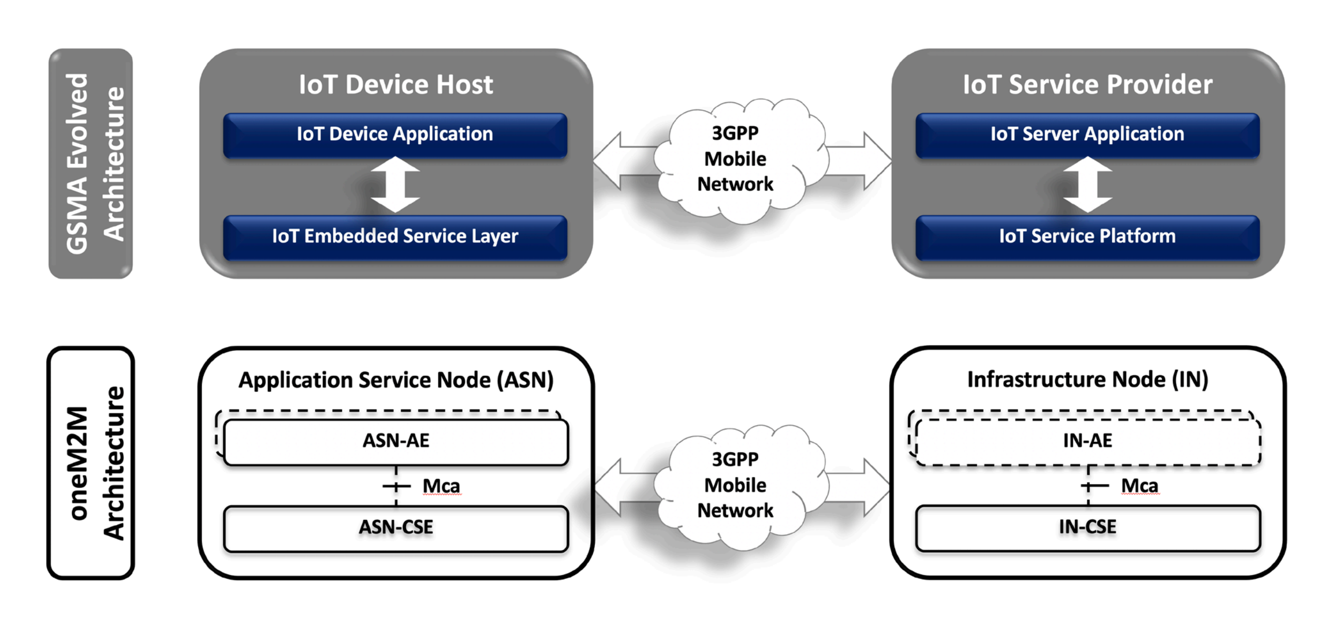


Figure 3- Evolved GSMA Architecture Compared to the oneM2M Standard Architecture

The oneM2M standard also defines Infrastructure Node Common Service Entity (IN-CSE), an IoT server element which can also include mobile core network interworking services. These services make use of APIs that are exposed by the 3GPP SCEF function [4]. oneM2M CSE entities can be configured to optimize device communication patterns whereas some of the ***TS.34 IoT Device Connection Efficiency Guidelines*** simply block or prevent access to the communication features of the radio chipset. Furthermore, using the oneM2M standard-based architecture, IoT device manufacturers only need to implement the main business logic of their application (ASN-AE). In turn, oneM2M ASN-CSE effectively abstracts the requirements related to the mobile network from the application and protects the mobile network from any intended or unintended effects of that application on the mobile network. A oneM2M standard implementation that adheres to ***TS.34 IoT Device Connection Efficiency Guidelines*** will be described in the Release 4 of ***oneM2M TS-0026, 3GPP Interworking*** [2]. Furthermore, the oneM2M Alliance is also in the process of developing a detailed Technical Report that outlines how oneM2M ASN-CSE can be used to protect the mobile core network.

V. oneM2M Common Services Functions that Support GSMA Guidelines

A variety of Common Service Functions (CSF)[3] are offered by the oneM2M standard for IoT devices and applications. This makes IoT product development cycles much easier. The following is a description of some of those CSFs and how they can be used to support GSMA guidelines described above.

The Communication Management and Delivery Handling (CMDH) CSF uses policies to manage the delivery of messages between IoT devices, gateways and servers. CMDH capabilities include buffering messages and the selection of the underlying communication technology (cellular, Wi-Fi, …). CMDH policies also allow the transmission of messages based on the priority and the type of message. CMDH policies can be dynamically updated to reflect conditions in the mobile network, such as high congestion. CMDH features are well suited to implement many of the GSMA requirements and protect the mobile networks from poorly designed IoT applications. Table 1 illustrates a few examples of how GSMA TS.34 requirements map to oneM2M CMDH capabilities, offering MNOs the ability to adapt network rules dynamically.

Table 1 - GSMA Requirements Support Using the oneM2M "Communication Management and Delivery Handling" Function

GSMA Req. ID	Description	oneM2M Support
TS.34_4.2_REQ_002	Data should be aggregated by the "IoT Embedded Service Layer" into as big a chunk as possible before being compressed and sent over the communications network.	oneM2M <cmdhNwAccessRule> resource allows specification of the minimum amount of data that needs to be aggregated before sending messages over the mobile network.
TS.34_4.2_REQ_003	If permissible for the IoT Service, the "IoT Embedded Service Layer" should avoid synchronized behavior with other IoT devices and employ a randomized pattern.	The oneM2M <cmdhNwAccessRule> resource allows specification of a number spreadingWaitTime (SWT), such that before accessing the underlying network, the CSE will wait for an additional amount of time randomly chosen between 0 and SWT.
TS.34_4.2_REQ_011	The "IoT Embedded Service Layer" should always be prepared to handle situations when communication requests fail. Communication retry mechanisms implemented within an IoT device can vary and will depend on importance and volume of downloaded data.	The oneM2M <cmdhNwAccessRule> resource allows specification of backOffParameters that define how communication retries shall be handled when initial attempts have failed.

IoT device communication patterns can be described such that the oneM2M CSE can help configure both 1) the 3GPP network with information to both optimize mobile network parameters [4] and 2) the IoT device with appropriate power saving functionalities, such as Power Saving Mode (PSM) or Enhanced Discontinuous Reception (eDRX). Based on these communication patterns, oneM2M CSE can ensure that the IoT device properly applies several additional GSM requirements discussed below.

Table 2 - GSMA Requirements Support Using the Core oneM2M Services

GSMA Req. ID	Description	oneM2M Support
TS.34_4.2_REQ_016	The "IoT Embedded Service Layer" should be designed to ensure the application's network communication activity is not concentrated during periods of high network utilization.	The oneM2M <schedule> resource defines the time periods when the IoT device can communicate via the mobile network.
TS.34_4.2_REQ_018	Each time there is a need to send data over the mobile network, the "IoT Embedded Service Layer" should take into account the information communicated by the IoT device application about the importance and urgency of the data.	The application can specify message categories that indicate that the requests shall be sent as soon as possible or that requests can be buffered and forwarded later.
TS.34_6.0_REQ_004	The "IoT Service Platform" should be aware of the state of the IoT device and only send 'wake up' triggers when the IoT device is known to be attached to the mobile network.	oneM2M uses the 3GPP SCEF monitoring event API for device reachability. Additionally, the <schedule> resource indicates when a device can receive a 'wake up' trigger.

Radio Policy Management (RPM) is another category of requirements specified by the GSMA. oneM2M standards define device management capabilities natively or through interworking with an external device management server (e.g. OMA DM, OMA LWM2M). oneM2M management object resources can be used to manage the RPM values on IoT devices following the process defined by the GSMA. The oneM2M device management API is the same for both native oneM2M IoT devices and IoT devices that use other device management technologies.

Table 3 – GSMA Requirements Support Using the oneM2M Device Management Services

GSMA Req. ID	Description	oneM2M Support
TS.34_7.1_REQ_002	The communications module shall allow the IoT device application to query for a report of the currently stored parameters <NFM Active> and <Start Timer Active> using an AT command.	ASN-CSE would exercise the communication module API to reflect exposed parameters in a oneM2M management object.

Harmful deployment scenarios described earlier could have been avoided if the capabilities provided by the oneM2M standard were used from the outset to support GSMA ***TS.34 IoT Device Connection Efficiency Guidelines***. First, messages from the mobile device would have been buffered by the ASN-CSE until the communication connection was re-established. Second, the reconnection attempts would be spaced out and communication retries would have been limited. Third, the IN-CSE would be able to trigger communications from the devices once the connection issues were resolved. Finally, depending on the duration of the communication outage with the IoT server and the memory available on the device, the IoT service could have been restored with little or no impact to other data or services.

VI. 'No Harm to the Network' Deployment of IoT Services

Deployment of a oneM2M standard-based IoT Service Layer allows for a variety of techniques to manage the mobile network communication polices and service provider business logic as illustrated in Figure 4.

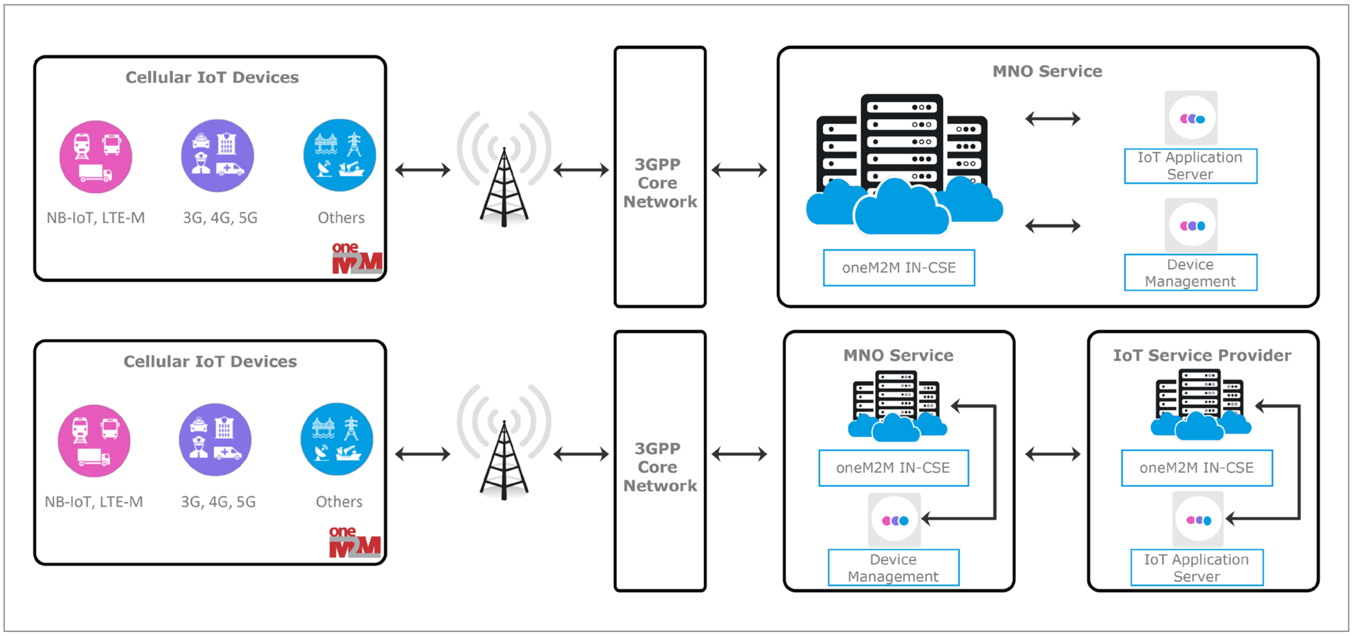


Figure 4 - Deployment of oneM2M Standard-Based IoT Solutions Within a Mobile Network

The oneM2M standard defines access control policies that allow the MNO to determine what visibility is permitted for the CMDH policies, device management and RPM details. Similarly, the IoT application data is protected by access control policies, so that a single oneM2M CSE can support multi-tenancy from many devices and many service providers. Furthermore, the oneM2M standard takes advantage of 3GPP SCEF APIs in the mobile network [4] to offer additional device performance and network optimizations. However, portions of the GSMA guidelines can be implemented and supported by oneM2M standard-based solutions before 3GPP SCEF capabilities become widely available in the mobile networks globally.

VII. IoT Services: Trust or Verify?

When MNOs are preparing to deploy large numbers of devices, they need to be certain those devices will operate properly and will not have any negative impacts on their overall networks and services. MNOs thus have test requirements that these devices must pass before their deployments. Since this process is repeated for each product that may be deployed on the mobile network, the complexity, time and costs of such deployment testing and device certification can run very high.

oneM2M Alliance defines and builds conformance test specifications that simplify this process, very similar to mobile handsets certification. oneM2M test cases are implemented in Testing and Test Control Notation version 3 (TTCN3) widely used in 3GPP testing. Furthermore, oneM2M standard certification has been managed by Global Certification Forum (GCF) since the beginning of 2019. Authorized test labs for oneM2M standard-based solutions are available in Asia, Europe and North America.

When IoT devices are built using a certified oneM2M standard-based solution for the IoT Embedded Service Layer, MNOs can be confident that the devices will not harm the network. The oneM2M conformance process helps ensure the reliability and scalability of oneM2M standard-based IoT products, both for IoT service providers and for the MNOs.

VIII. Conclusion

This article started with a summary of some of the real-world issues that mobile network operators have faced with IoT deployments on their mobile networks. GSMA has recommended guidelines for reducing the likelihood of these types of problems and more efficient cellular IoT connections. oneM2M standard-based solutions can enable granular control of devices through policies which can be dynamically adjusted to account for traffic and service needs. Using a GCF-certified oneM2M standard-based solutions enables MNOs to have more control over their networks. MNOs can trust that a oneM2M standard-based solution supports efficient communications and has been verified to pose ‘No Harm to the Network’. oneM2M also offers a more advanced approach to dealing with sub-optimal IoT application designs. Rather than block signaling storms with the limited functionalities provided by mechanisms such as Radio Policy Manager, oneM2M allows MNOs to optimize IoT application behavior. The possibility emerges to sell a true “managed” connectivity with Quality of Service to customers.

Since oneM2M is an open standard, all stakeholders are assured that there is no vendor lock-in as with many proprietary implementations. MNOs can be confident that their mobile networks are protected from harm and product manufacturers can realize faster time to market and lower costs. oneM2M standards-based solutions will ensure efficient and more resilient networks for many years to come.

REFERENCES

[1] GSMA TS.34 - IoT Device Connection Efficiency Guidelines V 5.0, 08 January 2018 https://www.gsma.com/newsroom/wp-content/uploads//TS.34_v5.0.pdf

[2] oneM2M TS-00026, 3GPP Interworking V3. www.onem2m.org/.

[3] oneM2M TS-0001, Functional Architecture V3. www.onem2m.org.

[4] Michael Starsinic, Dale Seed, and Chonggang Wang, "An Overview of 3GPP Exposed Services for IoT Service Platforms", ACM GetMobile, Volume 22, Issue 2, pp. 16-21, June 2018.

About the Authors

Bob Flynn (Bob.Flynn@chordant.io) is a Member of Technical Staff at Chordant, where he leads the architecture and design of the oneM2M component of Chordant® IoT Platform. Bob also represents Convida Wireless™, a partnership between InterDigital and Sony that is focused on IoT research and development, in oneM2M SDS and TDE working groups. See more: https://www.linkedin.com/in/bob-flynn-1738a45/.

Miguel Rodriguez (Miguel.Rodriguez@telekom.de) is a Senior Manager IoT Device Verification & Engineering, Deutsche Telekom. With 18 years of experience in the telecommunications industry, including network infrastructure, mobile phone, and operating system vendors, Miguel joined Deutsche Telekom's Integration and Validation organization in 2016 to build up its international IoT solution validation services and strategy. See more: https://www.linkedin.com/in/miguelrodriguez/.

About Chordant®

https://www.chordant.io/about/

About Deutsche Telekom

https://www.telekom.com/en/company/company-profile