GTI IoT Service Layer Architecture White Paper

http://www.gtigroup.org

IoT Service Layer Architecture

WHITE PAPER

V1.0

Version	V1.0
Deliverable Type	□Procedural Document √Working Document
Confidential Level	□ Open to GTI Operator Members □Open to GTI Partners √Open to Public
Program Name	GTI MIoT Program
Working Group	
Project Name	Open Platform Project
Source members	Qualcomm, Sprint, CMCC, Huawei, Ericsson
Support members	
Editor	
Last Edit Date	04-19-2018
Approval Date	

Confidentiality: The GTI documents may contain information that is confidential and access to the documents is restricted to the persons listed in the Confidential Level. This document may not be used, disclosed or reproduced, in whole or in part, without the prior written authorisation of GTI, and those so authorised may only use this document for the purpose consistent with the authorisation. GTI disclaims any liability for the accuracy or completeness or timeliness of the information contained in this document. The information contained in this document may be subject to change without prior notice.

Document History

Date	Meeting #	Version #	Revision Contents
2017-05-28		0.1	Skeleton Draft
2017-10-10		0.2	Initial Draft
2017-10-25		0.3	Initial Draft (Input combined – CMCC, Sprint, Qualcomm)
2017-10-25		0.31	Initial Draft (revised – Huawei)
2017-10-28		0.32	To revise the skeleton
2017-11-21		0.33	Incorporate Ericsson input of section 2
2017-12-09		0.34	Incorporate all inputs
2017-12-13		0.35	Add Executive Summary and editorial Changes
2017-12-18		0.36	Refining the wording
2017-12-19		0.37	Accepted all changes in v0.36, refine the formatting.
2017-12-29		0.5	The version is refined by Qualcomm based on 0.37 for further review. Need to further check all the marker /*OPP*/ and comments.
2017-12-29		0.51	Refined Chapter 1 and updated all references
2018-01-06		1.0	Accepted all changes and Update to 1.0 version for sending out.
2018-01-08		1.01	Change section 5.9 into “Interworking”
2018-01-09		1.02	Refined the format.
2018-04-19		1.03	Change the cover and incorporated the GTI review comments

Contents

Executive summary

No one doubts on the trend of the IoT. Bunches of devices around us already get connectable, and more and more devices are expected to be connected in future. It is so exciting that the connected devices will create new applications, bring human life convenience, improve human capabilities, increase productivity and spawn new businesses.

For mobile operators, the first thing drawing attentions is the connections from IoT. According to market forecast there will be 27 billion of connections by 2025[2]. Although the estimates vary slightly in different versions of forecast[3], the number of connected devices will be in tens of billion in next ten years. Operators revenue, in the telecommunication, comes not only from the number of connections, but also the traffic passed on the connections -- the number of minutes of voice or bits throughputs. In Internet era, although operators become sort of pipe for on top applications, operators do benefit from the fast expanding throughput requirement from the upper layer applications. But for IoT, when each sensor establishes a connection sending only a few bytes, it is wondering how operators will benefit from IoT.

To address this, GTI formed a project called **O**pen **P**latform **P**roject (OPP) at the beginning of 2017, and made a try to figure out the values that operator can get from the IoT. GTI OPP team firstly investigated some potential issues of present IoT and explored what and how operators can contribute to IoT.

Cellular industry usually touts features like ubiquitous coverage, manage QoS, and security as its main advantages. While as for today, most IoT applications run in a silo mode, data is collected and used by verticals themselves. Such silo mode will become barrier for new IoT applications. There would be trend that some regional IoT service providers ask for data in another geographic area[4], and some verticals ask for other verticals data for enhancing the analysis. IoT will benefit from big data. The secrete of big data is about what, not why, it need not to be aware what data should collect for a special purpose, rather, the big data let data speak to everyone[4]. As the IoT evolves, easily data sharing will be highly expected. Putting the data together and making them easily be exchanged will be a must in future.

More Importantly, the underlying mobile network is an existing large-scale platform for bearing the IoT services. Getting the underlying transport network capability to be exposed to the IoT applications in a simple way while offering additional and commonly needed functions and guaranteeing a robust protection of the network from inefficient usage will provide differentiated competition for mobile operators' IoT platforms versus other over-the-top offerings. IoT developers look forward to object-specific data that can represent things, especially the structured data that can be analysed by machine. This is quite different than bits packet services which operators provide now. It is highly expected that the things and their properties represented in object-specific data can to be easily accessed, analysed and shared.

From GTI OPP point of view to address above IoT requirement, the best thing operators can do for the diverse IoT applications, is to provider common service capabilities to vertical applications, such as data management, data repository, communication handling and object data delivery, application management, service discovery, location, device management, group management, notification and subscriptions, and others. These easily used common functions will work as a service layer to facilitate opening the operator’s capability to IoT applications. It would be hard for operators to provide end IoT applications, as operators do not possess the knowledges in a specific domain to drive the end-to-end IoT market.

Internet of thing doesn’t only mean interaction between things to platform and platform to things. Although the platform on the cloud side acts as an intelligent agent in the middle, the data come from IoT devices (sensors) would finally be used to trigger other IoT devices (actuators). IoT is about things to things, interoperability is not only necessary, but essential in this situation for large scale deployment.

LwM2M as one standardized solution which is capable of RESTful features has attract lots of attentions among mobile operators due to simple way to help collect object-orientated data, and has been deployed by several operators.

oneM2M which was designed in RESTFul style from very beginning is identified as a scalable, efficient and robust service layer solution. oneM2M enables a horizontal IoT service layer platform, regardless of existing sector or industry solutions. While essentially being independent of the underlying network technology, it can use MTC and eMTC optimizations of 3GPP-based networks to enhance efficiency and offer IoT/M2M related network functions. Especially with a policy-driven data delivery mechanism will help address the signalling storm issue of mobile network that caused by IoT.

We are looking forward the fragmented IoT platform technologies evolving to a unified service layer solution, and looking forward to that to help operators create new business, as well as help the world to eliminate the isolated information island.

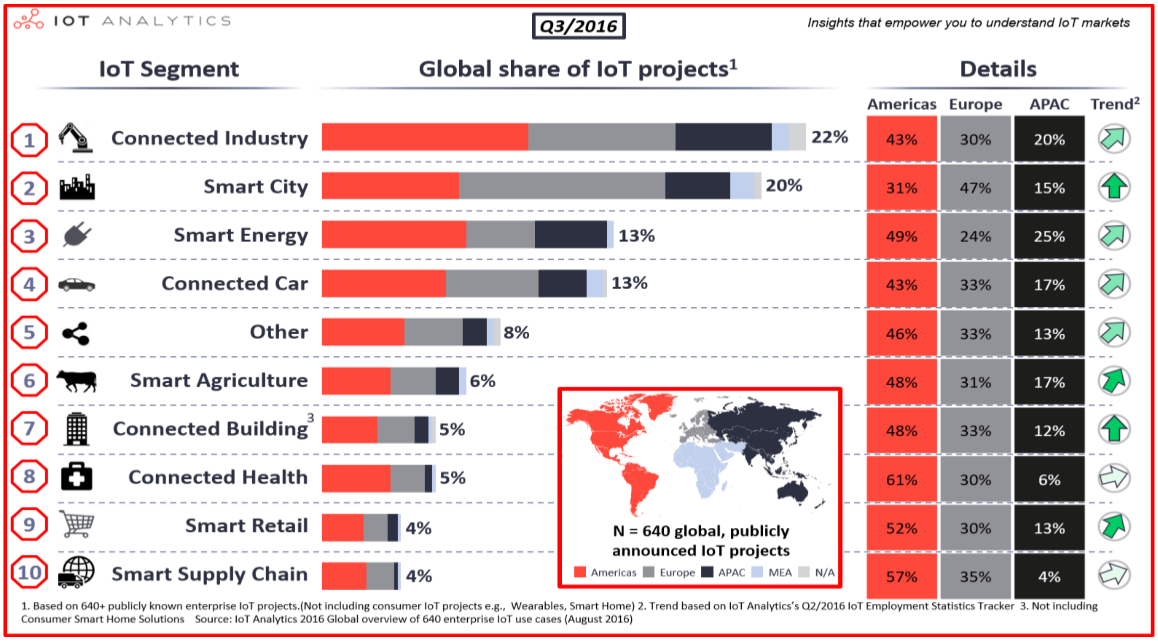
Terminology and Abbreviation

Term	Description
3GPP	3rd Generation Partnership Project
AE	Application Entity
ASN	Application Service Node
AND	Application Dedicated Node
ComSS	Communication Service Suite
CSE	Common Service Entity
CSF	Common Service Function
CMDH	Communication Management and Delivery Handling
CRUD	Create Retrieve Update Delete
HTTP	HyperText Transfer Protocol
IoT	Internet of Things
IN	Infrastructure Node
IPE	Interworking Proxy Application Entity
JSON	JavaScript Object Notation
LwM2M	Lightweight M2M
M2M	Machine to Machine
MAF	M2M Authentication Function
MN	Middle Node
MNO	Mobile Network Operator
MQTT	Message Queuing Telemetry Transport
MTC	Machine Type Communication
NoDN	Non-oneM2M Node
OPP	Open Platform
QoS	Quality of Service
SL	Service Layer
PII	Personal Identifiable Information
URI	Uniform Resource Identifier
WLAN	Wireless Local Area Network
XML	Extensible Markup Language
XSD	XML Schema Definition

1. IoT era is coming

IoT is a new business opportunity not only for various vertical industries, which intend to use the technology to increase productivity, reduce operational cost, and improve overall business efficiency, but also for mobile operators who have served M2M businesses for quite some time and have significant expertise on the design, deployment, and maintenance of wireless systems.

Applicability of IoT is generally bounded by imagination and creativity of human mind, but in general areas like smart cities, mobile health, smart utilities, environmental monitoring, asset tracking, and connectivity on a variety of levels as seen as the main applications of IoT at this stage. Figure 1 summarizes main IoT applications spanning all sectors 71[1].



1. Based on 640+ publicly known enterprise IoT projects.(Not including consumer IoT projects e.g., Wearables, Smart Home) 2. Trend based on IoT Analytics's Q2/2016 IoT Employment Statistics Tracker 3. Not including Consumer Smart Home Solutions Source: IoT Analytics 2016 Global overview of 640 enterprise IoT use cases (August 2016)

Figure 1 Main IoT applications

Gartner Group research suggests that the fastest growth areas to be smart buildings, both commercial and residential. In these environments, connected things will include temperature controls, LED lighting, healthcare monitors, smart locks, and sensors such as motion detectors and carbon monoxide alarms. Machina Research report suggests that there are 6 billion IoT connections in 2015, a number expected to grow to 27 billion by 2025, which indicates a compound annual growth rate of 16% [2]. On the other hand, Cisco reports that there were 5.8 billion M2M connections in 2016and there will be 13.7 billion IoT connections by 2021 [3].

In terms of IoT deployment geographies, US and China are expected to be the two dominant forces in the global IoT market, each counting around 20% of connections. IoT revenue opportunity is expected to be $3 trillion by 2025, according to Machina Research report. Of that figure, $1.3 trillion will be generated from end users (i.e. devices, application revenues, and connectivity) and the rest will come from IoT-related sources like application development, system integration, and data monetization.

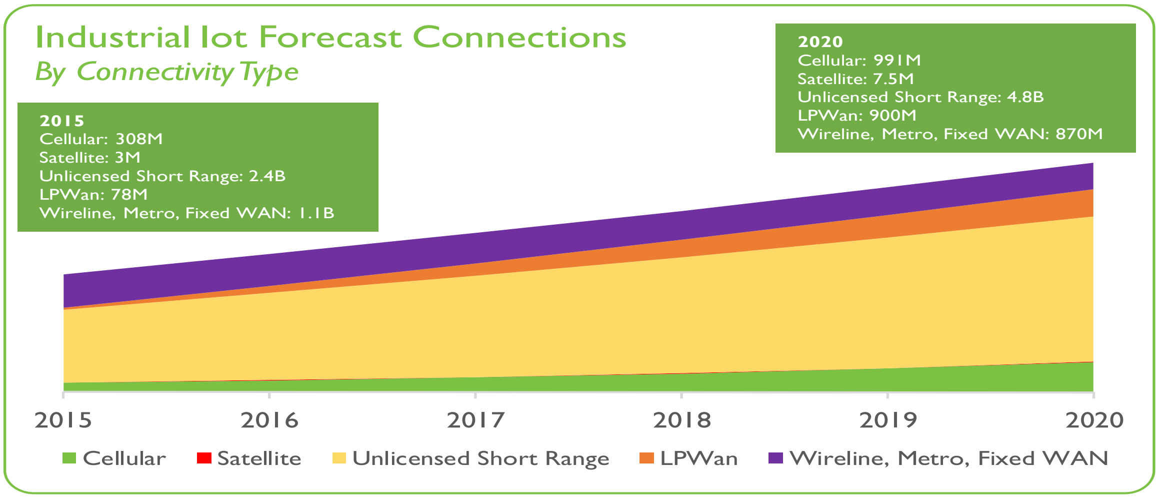


Figure 2 Industrial IoT Forecast

It is well understood that the overwhelming majority of IoT connections may not be provided by cellular IoT technologies. For instance, Machina Research estimates that 71% of all IoT connections rely on short range technologies such as Wi-Fi, Bluetooth, Thread, Zigbee, etc., mainly driven by IoT adoption in consumer electronics, building automation and security.

Based on a granular breakdown of the LPWA forecasts, Machina Research predicts that the two relevant 3GPP-defined standards, LTE Cat-M1 and LTE Cat-NB1 will collectively exceed 50% of the market within the next 5 years. The 3GPP-based technologies will come close to the non-3GPP LPWA networks already in 2021. In terms of absolute numbers, the 56% of the 2022 market equals to 862 million active connections, with the remaining 44% attributed to the non-3GPP technologies [6].

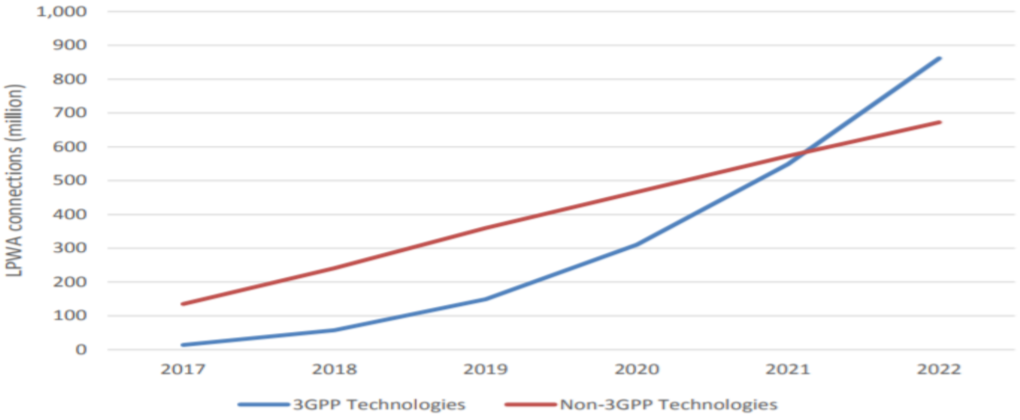


Figure 3 3GPP-Based LPWA networks will surpass dedicated technologies in 2022

2. Operators add values to IoT

The IoT business model for mobile operator is to discover who our customer is, to understand what the customer values, and how we deliver value at an appropriate cost with specific profit mechanism. We design business models and trigger virtuous cycles that expand both value creation and capture over time. The success or failure of any business model depends largely on how it interacts with models of other players in the value chain.

In the Internet of Human era, cellular voice and data applications are primarily serving human for our basic needs of communication, education and entertainment. Mobile network operator business model depends on not only the number of devices, but also the traffic volume goes on the connections -- the number of minutes of voice call makes and the number of gigabytes data application consumes. The cellular network becomes data pipe for the over-the-top applications and mobile network operators benefit from the fast expanding data consumption from these voice and data applications.

Moving forward the Internet of Things era, services are not only providing better way of life to individual but also assisting business and making a better society for mankind. The evolving of IoT service is all about improving efficiency in every aspect of mankind by making Things smarter.

Cellular industry usually touts features like ubiquitous coverage, manage QoS, and security as its main advantages, as shown on Figure 4.

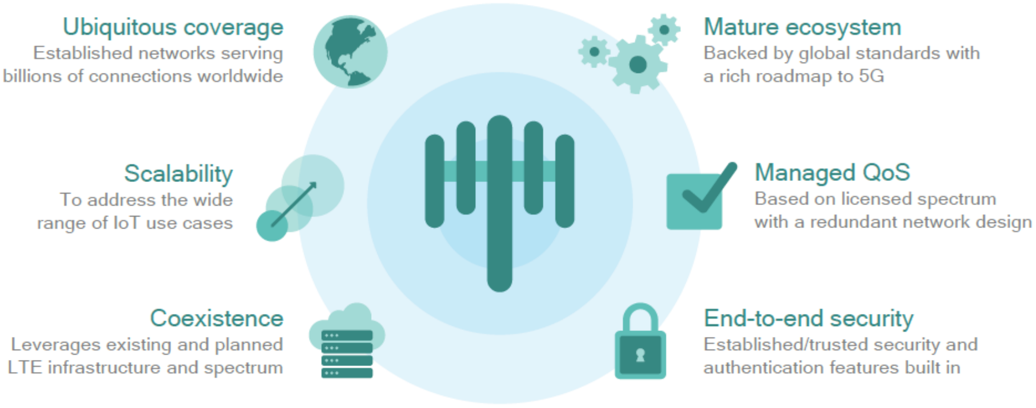

Ubiquitous coverage
Established networks serving billions of connections worldwide
Mature ecosystem
Backed by global standards with a rich roadmap to 5G
Scalability
To address the wide range of IoT use cases
Managed QoS
Based on licensed spectrum with a redundant network design
Coexistence
Leverages existing and planned LTE infrastructure and spectrum
End-to-end security
Established/trusted security and authentication features built in

Figure 4 Key benefits of Cellular technologies for IoT

The ubiquitous connections coverage is the pre-condition for IoT, which provides customer a quickest way to set up over-the-top IoT applications. IoT application is not just a connection between two nodes. Unlike the human communication service where human is the intelligence power to process the information transmitted over the connection and make value from such information, the IoT nodes need to be empowered by the data analysis capabilities from the IoT platform side.

Figure 5 shows an example of IoT applications where the sensors send data to the platform. On the platform side, besides the data collected from the sensors, the platform asks the weather information and other necessary data from the cloud. The platform gets the information processed and analyzed, figures out what request should be sent to the heater or air conditioner to control the environment without human intervention.

The platform is the intelligence engine for the IoT device. The IoT end nodes usually are resource constrained. They must leverage the platform's computing capability/intelligence to function in a coherent fashion.

To enable such IoT applications, the functionality of the IoT platform specified below are necessary:

- Preliminary Data processing –how to filter the data and visualize the data.
- Extensive data collecting - collection of the data from the sensors anywhere for further processing.
- Profession Services – the knowledge of the application and the data: what data should be collected and how to use the data to make a decision.

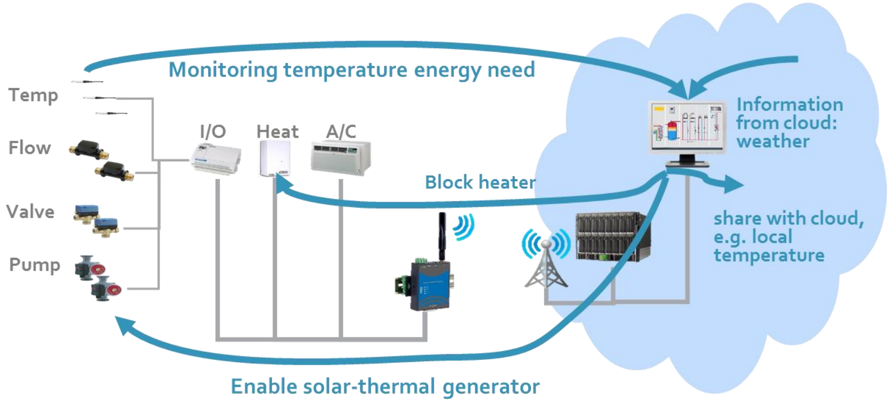

Monitoring temperature energy need
Temp
Flow
Valve
Pump
I/O
Heat
A/C
Information from cloud: weather
Block heater
share with cloud, e.g. local temperature
Enable solar-thermal generator

Figure 5 an IoT Application Example

It should also be mentioned that not all IoT connections will terminate directly to a macro deployed wireless network or cloud, but intermediate nodes like IoT routers/gateways may play a significant role on IoT overall network deployment topologies. It should also be mentioned that not all IoT connections will terminate directly to a macro deployed wireless network or cloud, but intermediate nodes. For instance, analyst firm Berg Insight reports that 2.7 million cellular M2M routers, gateways, and modems for connecting IoT devices were shipped in 2015, a number predicted to grow to 5.7 million units by 2020, again a compound annual growth rate of more than 16%.

2.1. Multi-layer multi-vertical network-sliced service

Data processing and data analysis for IoT applications could be multi-layer services. From operators' point of view, the service consists of multiple horizontal layers to different verticals. Figure 6 diagram presents one concept where services could be offered in each layer and within each vertical by slicing the IoT network to meet requirements from each layer-vertical combination.

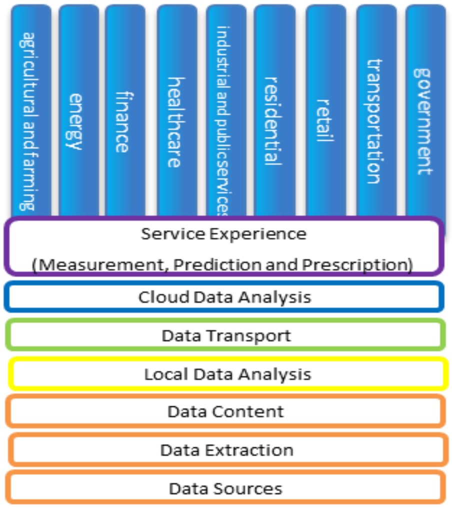


Figure 7 Main IoT Multi-Layer Multi-Vertical Network-Sliced Services

These components were added in to the overall architecture with the purpose of offering service layer intelligence to potential customers. Customers of this service may or may not be owners of Cellular IoT devices, or own the data. Some customers may only need special data analytics to be run and willing to pay for. A case where connected refrigerator offering food delivery to the door was showcased, but that's one of the more obvious use cases. The potential is not just limited to the connected device owners or manufacturers. For example, consider the case where financial institutions gain access to analytics to agricultural data that is down to the square feet from various farms across the nation. With the specific data analytics run on the data collected, it may be possible to estimate the prices of products and this is quite important for commodity trading.

The way to achieve such result can become possible with a IoT **S**ervice **L**ayer being offered as part of the Cellular IoT support. A business entity with Cellular IoT network offering can offer abstraction layer to the customers looking for off the shelf type IoT solutions, and store all the data in large database. Rapid technologic advances in data transfer, storage and processing and the availability of numerous connected devices offer a basis for building service-oriented business models. Big data suggests that sensors and connected devices are not limited to being a generator for tailored services. The challenge is to correlate the data gathered to identify potential cost savings in the business process and obtain better customer information and other competitive advantages that will capture value for the company.

A data analytics engine can be used in order to run analysis and generate automated reports for customers use. This analytics engine can be enhanced by allowing news and other external data to be consumed by the platform, only to offer meaningful suggestions that will help subscribers of the service to gain convenience, advantage, be more efficient and/or profitable.

2.2. Efficient data collecting and sharing

As mentioned that not all IoT connections will terminate directly to a macro deployed wireless network, intermediate nodes will involve different kinds of transport. This situation will make it very difficult to move the information bits across mix of different transports, especially to integrate the information bits from Things that reside in different networks and it is hard for IoT applications to deliver data consistently and efficiently.

Another difficulty for the IoT application developer is that object-specific data is often in non-self-explanatory form. There is requirement from IoT applications itself to represent the data in a format that the computer can deal with, and therefore the information and knowledge carried in the data can be visualized and to power the IoT applications.

As showed in Figure 8, most IoT applications is run in a silo mode, data was collected and consumed by verticals themselves. Some verticals can quickly create relative large-scale use cases, while compared to the real market size of IoT, their current market size is still small.

Such Silo mode will become a barrier for new IoT applications since some regional IoT service providers would ask for more data in another geographic area, and some verticals would ask for data from other verticals for enhancing the data analysis.

More important, big data is about what, not why, we don’t always need to know the cause of a phenomenon. We need not to be aware what data we should collect for a special purpose. The big data will let data speak to us [4].

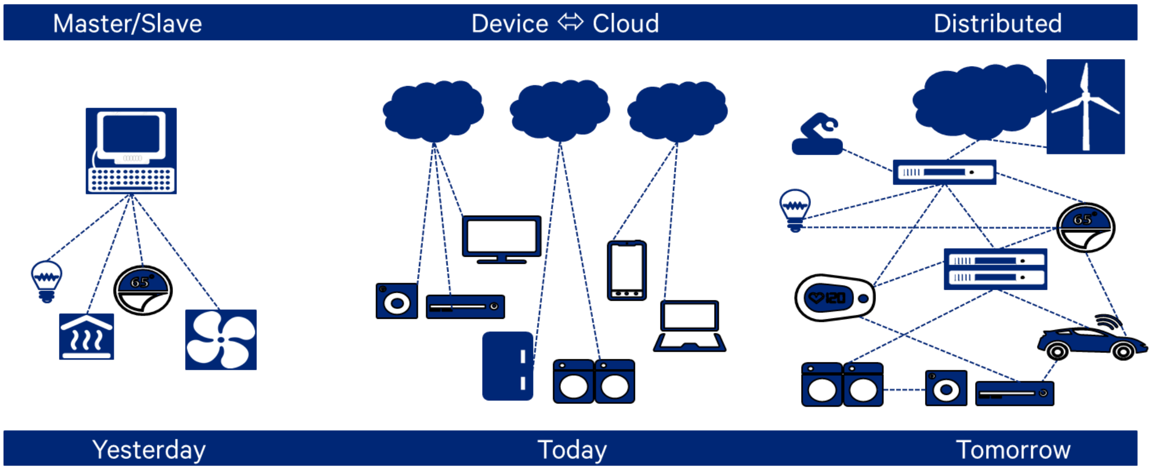


Figure 8 Horizontal Platform for Easily Data Sharing

Operators will continue providing the ubiquitous coverage of interoperable network for lots of verticals, and it is foreseen that more verticals will be served by operators in the future. IoT application developers are facing with problems of hard to use bits from devices and encounters with the problems of interoperability. Operators are being faced with signalling storm caused by IoT. As the experts of the complicated transport network and the complicated communication technologies, operators would be a key to enable efficient data sharing and drive maturity of IoT applications. Towards IoT, 3GPP is working on ultra-low latency (URLLC), massive MTC (mMTC) solution in 5G. And oneM2M is working on simple and easy to use RESTful interface with policy driven data delivery management.

2.3. Professional service

In new data from ABI Research's IoT Market Tracker, professional services are driving over 40% of global IoT revenues today. Professional services consist of a myriad of service activities including mobile, web and server-side application development, IT and OT systems integration, and consulting. However, by 2019, software platforms and analytics services will each drive more revenues globally than professional services in IoT markets.

2.4. Software platforms and analytics services

IoT software platform suppliers provide tools and services to greatly simplify extracting data from machines and things, managing them and building IoT applications. This group consists of a well-established set of suppliers who have focused their platforms on vertical markets led by vendors serving the OEM and fleet telematics segments. But it also consists of a long tail of platform vendors offering everything from very specialized services to a whole software stack including applications development. ABI Research believes that "verticalized" platforms will continue to hold the largest share of the IoT software platform market with new growth provided by the smart home and healthcare segments.

3. IoT service system of operators

In the past, telecoms operators treated their network providers as little more than “dumb pipes”. With the IoT revolution, operators want to play a new role to create new value. From estimates produced by Analysys Mason [7], connectivity is just a component in the whole generic value chain for IoT services. If operators take on a wider role in IoT, for example by providing a complete solution that incorporates device, application, service provision and integration as well as connectivity, they can earn a larger share of spend.

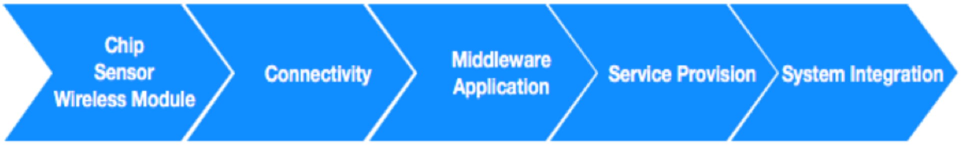


Figure 9 Generic Value Chain for IoT Services

It’s obvious that telecoms operators need to take up a different position with the IoT revolution. If telecom operators only sell connectivity services just like the past, they will put themselves in a weak position. Thanks to the central role of communications in many IoT deployments, how companies create value is often a function of the interaction between sensor technology and the network layer.

According to the business of IoT, operators have four possible approaches to IoT:

- **Connectivity**. This forms the basis of most operator IoT solutions and operators must ensure that they are well-placed to provide a range of connectivity options, such as NB-IoT.
- **Generic platform**. An operator provides basic tools and capabilities (such as device management) that developers can use to create IoT solutions. Focusing on the capabilities that are common across multiple applications should be a good choice.
- **Vertical-specific platform**. An operator offers platforms or capabilities tailored to a specific vertical market, such as healthcare. When selecting vertical markets, telecoms operators should avoid competition with tech giants.
- **End-to-end solution.** An operator offers all components of a solution. It may be justified for the largest vertical markets.

In these fours approaches, providing more than connectivity is not as straightforward as it may seem. Compared to vertical-specific platform provider and end-to end solution provider, connectivity provider and generic platform provider are low-to-medium risk roles for telecoms operators. When telecoms operators play as generic platform providers, they can gain higher value, but will also be subject to intense competition. The opportunities for operators are also for other players, who may have more vertical-specific experience or unique assets. Operators must play to the strengths of networks and focus on areas where their offerings have strongest differentiators. And in IoT the services providing by operators should be scalable and reusable without getting into the detail of specific vertical market solutions.

3.1. Architecture requirement

As mentioned earlier, operators could concentrate on what common elements it can provide for multiple vertical markets. These could include traditional operator strengths, such as connectivity, but also reach into different areas, such as hosting, support and application enablement. Bringing together existing capabilities, adding some new ones and providing a horizontal menu of capabilities for operators' own and partner solutions may be a good method.

Providing a platform is a frequently used way to provide capabilities and common elements. It is not new and has been implemented by many companies and others. For example, Amazon has a mix of revenue streams selling its own products to direct customers, selling third party products, and even selling full e-commerce solutions to others, by providing an E-commerce platform. Just like the E-commerce field, telecoms operators can also apply this approach to make a contribution to IoT.

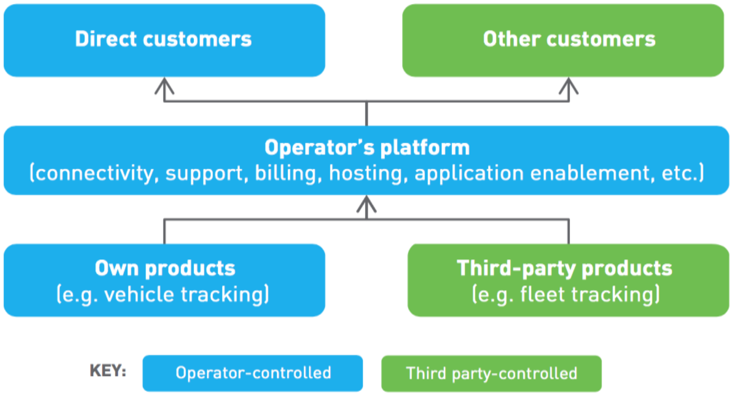


Figure 10 The Basic Model for Telecoms Operators Providing IoT Services 71[8]

The benefits of this model are that the operators gain access to customers controlled by third party. The operators can focus on the aspects where scale matters without getting into the detail of specific vertical market solutions.

In the basic model, the relationships among operator's platform, services(products) and customers have been shown. To build such IoT business, we need establish typical service architecture which is feasible and high-performance. For different companies and different subdivision area, service architectures will be varied in the practice. Even so, we can also describe it with a generic three-domain architecture as showed in Figure 11 [9].

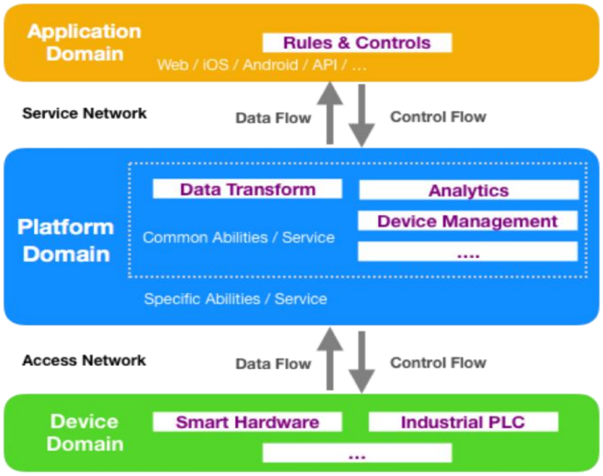


Figure 11 Generic Three-domain Architecture

The three-domain architecture consists of device, platform and application domains connected by networks. The networks in this as well as in the other architectures that follow all typically use a combination of enabling wireless and/or wired technologies such as RFID, Bluetooth, Cellular, ZigBee, Z-Wave, Thread, and Ethernet. As shown in Figure 11, the edge tier uses the access network to collect data from device domain. This data is forwarded over the access network to the platform, which processes data from the device domain for forwarding to the application domain, as well as processing and relaying control commands from the application back down to the device. The platform domain uses the service network to communicate with application domain, which provides end user interfaces, control commands and domain-specific applications.

In the three-domain architecture, the platform domain plays a key role. The platform should provide common abilities for IoT such as “data storage”, “time synchronization”, “device management” and so on. Moreover, some specific service can also become a part of it which bring more features and competitive power. Besides, big data analysis can make data more sense as so much data goes through the platform.

Figure 12 shows an example of open IoT platform architecture. The bottom tier is used for accessing the device and the top tier is used for supporting applications. In the middle tier, the IoT platform provides normal abilities (Common IaaS/PaaS) as well as in the other field’s platform and some special abilities for IoT (IoT BaaS/PaaS).

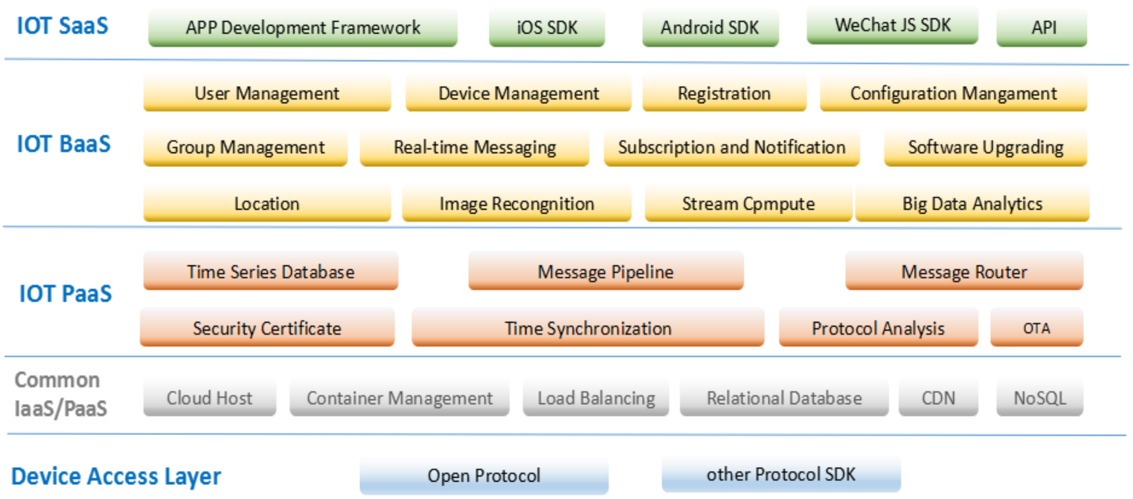


Figure 12 An Example of Open IoT Platform Architecture

3.2. Functional requirement

In order to define properly the IoT service system, it is also important to fully comprehend the typical characteristic features of IoT. Regardless of the role of telecoms operators, the key features of IoT systems must be considered.

- **Data correlation and information retrieval:** Smart data processing is a key IoT feature. The ability of today's IoT to distribute sensors widely and collect data quickly and effectively facilitates new forms of collaboration. Today's IoT also uses semantic modelling of the data it produces to make using the data more practical and intuitive, and to facilitate interoperation.
- **Communication**. Existing applications rely on communication that takes place between the edge and centralized servers, services and aggregation points. While factory integrated solutions only allow for applications and improvements within a particular manufacturing area, IoT platforms permit the collection of information from multiple heterogeneous entities and support collaborations beyond traditional enterprise silos.
- **Integration and interoperation.** Today's IoT solutions are characterized by varying degrees of integration and interoperation. Integration efforts often involve making systems work together that were not initially designed for interoperation. However, while not yet holistically available, within companies there can be a certain degree of integration and interoperability between products, where upper-level technologies are integrated with the technologies below them.
- **Security, privacy and trust.** As IoT is a dynamic system of systems, measures to attest the trustworthiness of IoT components throughout their lifetime are required.

As the special analysis for operators previously, communications service providers have multiple elements that could be included in a broader platform. The following functions could form part of an operator's broad menu' of IoT capabilities:

- **Connectivity:** an operator should provide customers with connectivity, even if this connectivity is not using its own network. This will involve roaming agreements for other countries, fixed connectivity, satellite connectivity and may even involve taking connectivity from other wireless networks (e.g. from a third party's LPWA network).
- **Billing and support**: few other organizations have the mechanisms to bill and support millions of customers. Telecoms operators do, and can provide it as a service to others.
- **Hosting:** for supporting internal products, but also for IT services for external clients, telecoms operators typically have environments suitable for hosting applications, including IoT/M2M applications. The operator's hosting environment may be especially important for applications that require data to be stored locally, or where latency could be an issue.
- **Application enablement:** many operators are assessing application enablement providers, some already have deals (e.g. Elisa with ThingWorx, Deutsche Telekom with Cumulocity) or have developed platforms internally. Essentially, this fits with the model of strengthening the platform – using the operator's scale to provide a more complete solution.

- **Professional services**: these are not scalable and reusable, and so do not formally qualify as a platform capability. However, for those with professional services, typically large operators with a strong enterprise focus, systems integration and even managed services could be included as part of the list of capabilities to be offered to potential M2M clients.

3.3. Building the open IoT service system

As can be seen from the previous chapter, the IoT market capacity will be huge. Compared with 10 billion traditional mobiles, PC, tablets and 4.6 billion IoT devices in 2015, IoT market will have a very fast development. Meanwhile the cellular IoT market is based on traditional telecom equipment market, it has original character of transition from monopolistic competitive market oligopoly. Entities' profit should be higher than it in monopolistic competitive market stage. Versus public cloud, IoT cloud part is closer to monopolistic competition stage. IoT sensor and UE market is at perfectly competition stage. As a result, it can be easy to understand, cellular IoT market has the transition character of from oligopoly to monopolistic competition.

IoT network operator is responsible for IoT network setup and its operating while the service provider of IoT cloud platform can provide devices access, security authentication, data storage and processing competence for specific APPs. The core competitiveness of IoT cloud provider is powerful cloud compute capacity. IoT platforms managing devices, connectivity, applications, and back-end integrations to legacy enterprise systems may ultimately become standardized, leading to a change in business models. There are various IoT platform providers around the world.

IoT platforms sit in the middle of this vast ecosystem, providing the middleware between the IoT endpoints and the repositories where the data collected from the endpoints will eventually reside. There are consumer-focused platforms, horizontally focused enterprise platforms, and industry-specific platforms. Among those providers, the leaders are IBM, PTC, AWS, GE Digitals and SAP whereas none of the carriers are even the major players according to IDC MarketScape report [10].

To be competitive, the operators must take some open strategies such as the horizontal strategies which target the existing and prospective rivals, allow the rival platform's users to interact with the focal platform's users, enable additional parties to participate directly in the focal platform's commercialization and support additional parties to participate directly in the focal platform's technical development. The conditions under those 3 strategies for opening mature platform include following [11]:

- **Interoperability**

Interoperability means the cross-platform transactions between their respective users or connected devices. For instance, CMCC's OneNet platform could be interoperable to PTC's ThingWorx. That would allow their subscribers or devices to exchange messages and enable one platform's users to interact with others, including supply-side users who offer complements.

- **Licensing New Providers**

At the beginning when the market is young, a sole platform may satisfy certain market requirements. As the market grows and matures, user segments with differentiated needs usually emerge. A single company may be unable to create a sufficiently broad array of features or products to answer the increasingly diverse needs. Licensing new providers is most attractive when new providers can offer innovative versions of platform products, rather than implementing all by the operators themselves. Competition with rival platforms may encourage a focal platform's sponsor to license additional providers with the goal of harnessing network effects and attracting additional users

- **Broadening Sponsorship**

To be more racial, the approach to invite other parties to jointly develop the core technology of the opening platform would bring more advantages such as decreasing the R&D costs by sharing those costs with other sponsors, a common standard which would be achieved from the incorporation among the sponsors and survive as the fittest proposals and finally the higher quality products could be created though the open processes for jointly developing technologies.

Another reason to be open in the IoT world is that the IoT journey has started from the phase of Connected Things to that of Connected Service towards that of Connected Ecosystem. The Connected Ecosystem is about the inter-connect of silo business processes, the horizontal integration across different industries and the cross-sector integration, according to Cisco.

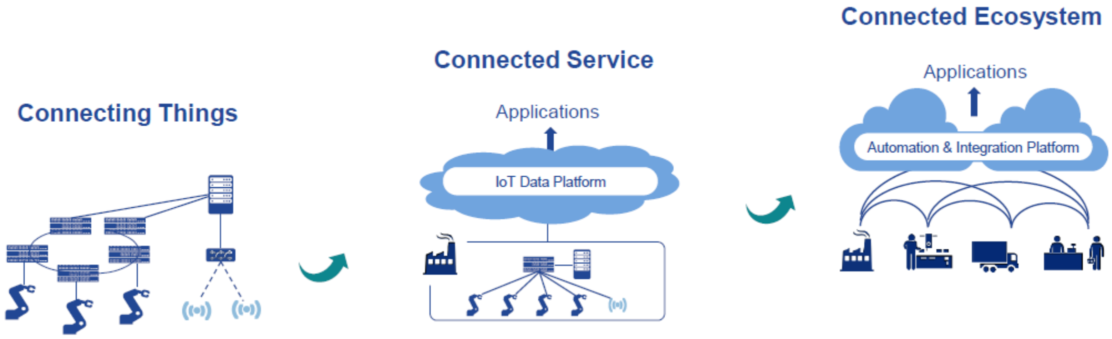


Figure 13 Open for Connected Ecosystem of IoT

In the third phase, the IoT platform must be open to enable the interoperability of different sectors, different industries, different geographies, and different ecosystem partners. IoT platform standards will begin to mature, allowing customers and businesses to obtain standardized access to devices and device data. During this phase, the emphasis and importance of physical devices will be reduced; instead, common device registries, interchangeable data-processing algorithms, and the co-alignment of artificial and business intelligence will become a reality.

Future IoT platforms will be recognizable from earlier IT and IoT solutions based on their open and flexible architectures. Earlier IT, M2M and IoT architectures reflected robust and closed-loop information systems. To achieve the benefits in IoT, open architectures have become one of the leading characteristics, together with the scalability, the flexibility, and the agility.

4. Standardization to boom IoT

4.1. Why standardized solution is necessary

The global IoT market growth faces the challenges due to the fragmentation of IoT technology landscape and the absence of established standards to enable the interoperability among IoT devices, applications, data collection/storage across geographies and across industry sectors. A common and open platform, which could gather different vertical IoT service providers and many other stakeholders of the IoT ecosystem, is an emerging need for cross industry interoperability rather than the industry-specific approach.

In the coming years, the dramatical and massive growth of IoT devices and application which would lead unbelievable traffic and usage data must be anticipated and considered if the solutions are to be ready in time. A common set of service layers, open interfaces and APIs of a standardised architecture is a reasonable approach to reduce investment, improve time-to-market and build a solid IoT business case. The use case of the smart meters is a good example. The utility service providers expect a stable network interfaces and manageable and upgradeable device management service to safeguard the investment for a long term, e.g. 20 to 30 years. The standardisation of an open IoT platform with the standardised architecture will easily bridge all the ecosystem partners e.g. the component providers, application developers, system integrators and wireless and wireline connectivity providers together to form such a solution. The elements of interoperability, portability, extensibility, agility and flexibility, as well as all common technical features will be embedded into the platform. To deliver those elements that the market requires, the standardisation, which improves the functionality-cost-quality trade-off and enables the fast pace for the new devices and applications, has been born and it is growing.

The standardization framework for IoT will solve the interoperability across networks, solve the interoperability of data, shape the device management and service automation capabilities and address the security concerns.

4.2. Standardized platform

In looking at the IoT standardization landscape, it becomes clear there is significant fragmentation of effort and overlapping of initiatives. The fragmentation is detrimental to achieving the smart and secure IoT ecosystem. The IoT industry will not take off without significant consolidation and the IoT industry will not benefit from economies of scale that standardization can bring.

- **Focus on solving business problems, not reinventing the wheel.**

The Internet of Things brings together so many previously disparate strands from technology and different industries. There are common requirements for connectivity, security and data handling that cut cross all businesses. The reality is that same services are developed again and again. A standardised architecture with a common set of service layer capabilities and open interfaces and APIs should also help M2M and IoT providers to reduce investments, time-to-market, development and on-boarding costs, and facilitate management of devices and applications. [1] When building a IoT platform becomes easy and feasible, providers can take more time and energy on business. This will help to build a solid M2M and IoT business case that relies on very small revenues, and even smaller margins [12].

- **Not all of the M2M and IoT providers have the ability to develop an IoT platform**

To accomplish a IoT platform, it requires not only the proficient skills of hardware design and embedded devices coding, but also the reliable supports from cloud servers and web UI. No one could be good at all these areas in the same time. It's hard to provide a complete IoT open source resource including hardware, firmware, server and web UI. For start-ups and non-tech companies, developing a proprietary IoT platform is a challenging project and it's obviously not necessary. In order to attract more participants, we need an easy approach to get into the field of IoT. The open and standardized platform will be a feasible solution for regular customers.

- **IoT standards ensure the interoperability for a large-scale ecosystem**

Providing a unique experience will be key for making the IoT a success – but many challenges exist [12]. If devices and applications can be abstracted from the underlying access networks and technologies, the objective “any app, any device, any network” will be achieved. Then there will be a large-scale ecosystem. However, the situation is usually that the highly fragmented market with limited vendor-specific applications.

It would be much nicer to have a single platform that could monitor all of your assets and give you consolidated map views of their location and alarm views of any issues. We need simple steps to easily and quickly control a variety of devices through the IoT platform. However, it may not even be feasible in theory that there is only a single platform in the whole IoT industry. We need standards enable easy integration across IoT platforms and application domains to ensure seamless interaction between heterogeneous applications and devices.

- **Proprietary platforms create information islands. Standardized platforms bring the real value.**

When building IoT applications, the cloud part where all information from sensors stored is important. Applications and services often need data at a higher level than the raw data provided by sensors. Moreover, data needs to be interpreted in the context of other sources of information. Standardized platforms stimulate large scale multi-vendor ecosystem with transparent product features and benchmarks, encourages industry investment, and promotes new business models. Standardized Horizontal Service Platform is key enabler for M2M and IoT provider, especially telecoms operators.

4.3. Standardized data model

Image that billions or even trillions of fragmented devices grow with heterogeneous middleware and applications, the IoT domains would be disparate and IoT-powered applications could not know how to validate, map, transform, correlate, and process that information without a standard data model. A common data model is a crucial element of the semantic interoperability which allows IoT cloud enabling applications to grabble the precise meaning of each piece of data that imported, acquired, retrieved, and otherwise received from elsewhere.

Among the standards defined by different standard organizations e.g. GSMA, OMA, ITU and etc. IPSO Smart Objects, initiated by the IPSO Appliance, provide a common design pattern, an object model, to provide high level interoperability between Smart Object devices and connected software applications on other devices and services. The group defines IPSO smart objects that conform to a network/device-agnostic data model that uses data objects to represent common IoT sensors. There is an object ID of each Smart Object and it represents a physical sensor, actuator, connected object or other data source. The reusable resources, which make up the smart object, represent static and dynamic properties of the connected physical object and the embedded software contained. With this standard data model, LWM2M based IoT devices or things and devices are able to communicate with each other by a “common language”.

Standard data model is also important to oneM2M Service Layer. Working on the abstraction of data heterogeneity and proving common tools is ongoing. Besides, oneM2M is exploring the new area of data semantics to improve data interoperability. In Release One, it has introduced a semantics add-on which can be used to discover the attributes of a platform with the names and locations of resources [12]. However, operators are not leading in the standardization although they are members of OneM2M and other standard organization. It is valuable to raise operators voice and have more support on open data model for IoT applications from operators’ open platform point of view.

Apart from the IPSO data model, data interoperability also is a technical barrier that prohibits the realisation of the full potential value of IoT Big Data. IoT Big Data Harmonised Data Model, initiated by GSMA, is a solution to address that problem. The document of” IoT Big Data API

Directory” provides a framework for the delivery of IoT big data services that recognises the many different approaches towards the services that are offered and the technology choices that are made. The proposed architecture offers a degree of flexibility which allows IoT big data services to be offered in multiple ways. That model mainly focuses on Agriculture, Automotive, Environment, Industry, Smart City and Smart Home six verticals [13].

IoT Big Data Harmonised Data Model is a common approach to data sharing, lowers costs and creates opportunities for IoT developers, data brokers and data providers. Operators are key participants in the delivery of an IoT big data ecosystem, although much of the IoT data that is collected will come from a range of data provider partners. In summary, IPSO and IoT Big Data Harmonised Data Model are two well-known standardized data models which target to solve data interoperability and recognize the full potential value of IoT Big Data.

4.4. On the way to standardization

“In October 2009, the first IoT Architecture (IoT-A) stakeholder workshop was held in Paris, France. Through a series of interactions and developments of an Architectural Reference Model for the IoT, the IoT-A Reference Architecture was finally defined in 2012. Following IoT-A and other Seventh Framework Programme for Research and Technological Development (FP7) projects such as Sensei and Future Internet Initiative in Europe (Fi-Ware), the main work-around architectures have moved to the European Telecommunications Standards Institute's (ETSI's) Technical Committee for Machine-to-Machine (TC M2M) communications and finally to oneM2M, publishing Release 1 in January 2015 [14]. oneM2M, now, becomes the leading global standardisation body for M2M and IoT with the purpose to develop a single horizontal platform for the exchange and sharing of data among all applications.

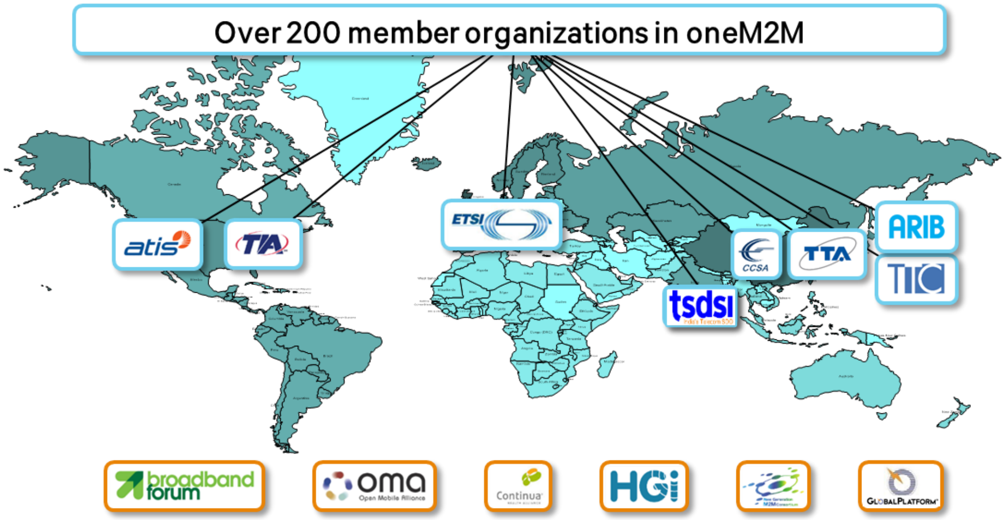


Figure 14: oneM2M Global Standardization Organization for IoT Platform

oneM2M has released a set of specifications to build platforms for the broad industry IoT solution. The specifications define the technology or approach of how to integrate the data and services across different organisations in various sectors and geographies. A framework for interworking with various technologies and a distributed software layer have been defined by oneM2M to ensure the re-use of current ICT and IoT technologies as much as possible.

5. oneM2M open IoT service layer

IoT vertical applications are keep emerging. The IoT end nodes usually resource constrained device, where the platform on the cloud side plays significant important role for IoT system. The cloud side platform seems an intelligence engine for IoT applications and IoT end nodes would leverage the platform's computing capability/intelligence to work automatically and autonomously.

In recent years, lots of IoT platform emerge, they usually provide some service enablement functions for vertical applications, such functions involve data delivery, data management, data accessing, device management as well as authentication etc. With these common functions, vertical application developers can focus on developing the specific features for specific applications.

The powerful "IoT platform" usually requires knowledges across different domains, it's intelligence would come from correlated data processing and analyzing with expertise in a specific domain.

Although it is impossible for mobile operators to possess all the diverse knowledge to drive the IoT applications. The IoT applications do need operators to help on collecting extensive data that can be processed by the computer easily and efficiently. As communication technologies have been developed for several generations, even the simplest IoT application would involve different kinds of transports. The involved transports make it very difficult to move the information bits across mix of different transports, especially to integrate the information bits from things that reside in different networks, and it is always hard for IoT application to deliver data consistently and efficiently. A middle layer software would be a great value for the IoT developers.

And the data which previously are not thought related to a specific domain or an IoT applications, might become relevant on some day, and will help to expose new 'truth'. The correlations of data may not tell us precisely why something is happening, but they alert us that it is happening, that is the 'big data' brings to us. We don't always need to know the cause of a phenomenon; rather, we can let data speak for itself [5]. This will require the data from different verticals to be easily shared between different verticals. On one hand, it asks for a middle layer on the cloud side platform to take care of the data, on another hand, the big data itself even require the data can be easily shared between different cloud side platform, which calls for the interoperability of the middle layer of the platform.

This is middle layer software is the **IoT Service Layer** which we think would be the enabler for large scale IoT services.

And we should always keep aware that internet of thing doesn't only mean interaction between things to platform and platform to things. Although the platform on the cloud side acts as an intelligent agent in the middle, the data come from IoT devices (sensors) would finally be used to trigger other IoT devices (actuators). IoT is about things to things, interoperability is not only necessary, but essential in this situation for large scale deployment.

Formed in 2012 by seven of the world's preeminent standards development organizations, oneM2M is developing a unifying **Service Layer** for the exchange and sharing of data among all applications. This Service Layer is a horizontal platform that can be implemented in the cloud side IoT platform, in the middle nodes of the IoT network, and the end-node of the network. The Service Layer provides common functions support to IoT developers for quickly development of IoT applications with lower CAPEX and OPEX. oneM2M as an open standardized **Service Layer** solution of common capabilities for IoT, addresses the fragmentation of the IoT platforms. oneM2M established through an alliance of standards organizations to develop a single horizontal platform for the exchange and sharing of data among all applications[12].

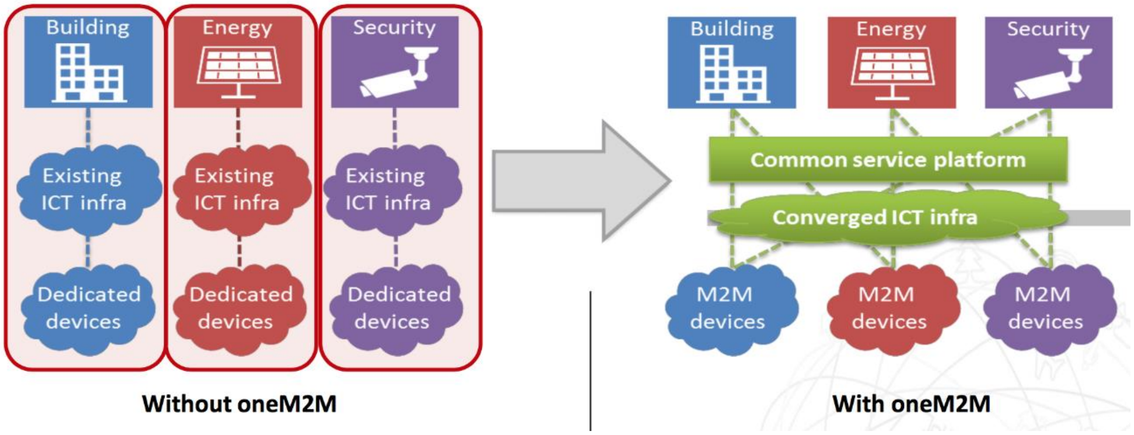


Figure 15: IoT Cross-Domain Interoperability [12]

5.1. oneM2M service layer architecture

oneM2M models the internet of thing system as a three-layers system, including application layer, common service layer and communication network layer. The Service Layer specified by oneM2M is an end to end IoT service layer solution for IoT applications, which can be used to build an end to end Service Layer for IoT service operators. The Service Layer can reside in the IoT end nodes, IoT gateways, as well as the cloud side infrastructures – IoT Service Platform, to constitute an end to end system for supporting IoT applications of different verticals and accelerating new IoT applications development.

The Service Layer for IoT network system consists of CSEs embedded in the nodes which are connected with each other. The CSE defined by oneM2M is a middleware in a IoT node. It talks to the upper applications via Mca interface in the northbound, and makes use of the underlying communication network service via Mcn interface in the southbound. Mcc is the interface between CSEs of different nodes in the east/westbound. An IoT node can use another node's capabilities via the Mcc interface. Mcc interface enables the CSEs as the middleware in different nodes building up a comprehensive service layer for IoT network.

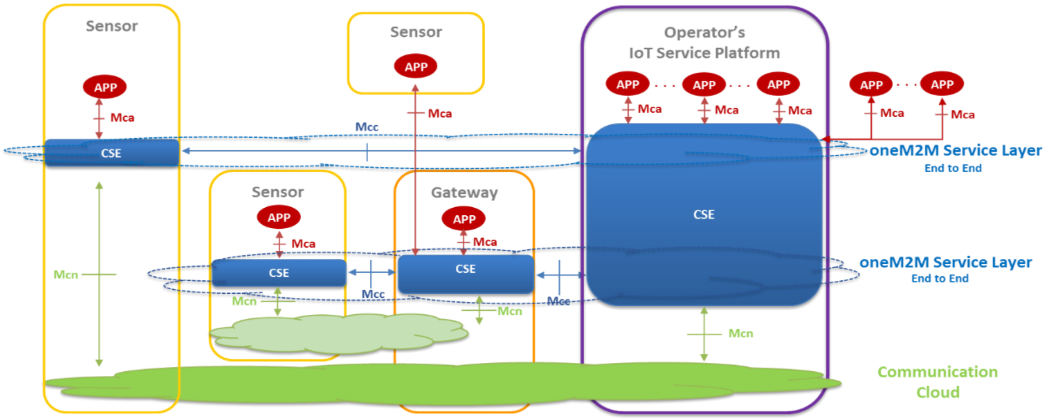


Figure 16: oneM2M Service Layer Architecture

In oneM2M, the App in Figure 16 using the service/capability provided by CSE is referred as AE, and the communication network which provides communication service and possibly other service is referred as NSE.

There are four types of nodes defined in oneM2M depending on the existence of AE and CSE as well as the position in the architecture configuration.

- Application Dedicated Node (ADN): a leaf node which has at least one AE but no CSE
- Application Service Node (ASN): a leaf node which has at least one AE and a CSE
- Middle Node (MN): a node has a CSE and zero or more AEs, sits between ADN/ASN and IN
- Infrastructure Node (IN): a node in the cloud as the platform which has a CSE and zero or more AEs

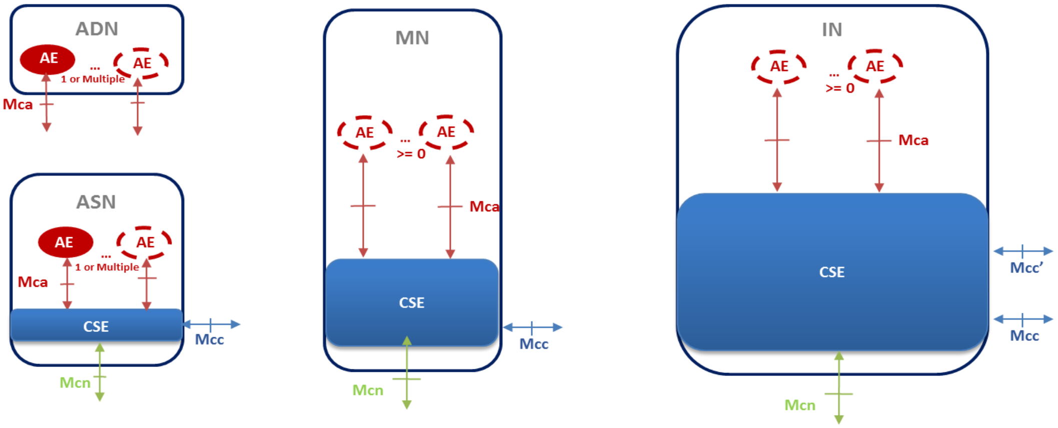


Figure 17: oneM2M Nodes Configuration

Considering the huge number of Internet of Thing nodes, Internet of things application diversity, and regional operation needs, it would be impossible that a single Internet of Things platform provider is capable to support and operate all nodes and applications. So oneM2M defines the Service Provider (SP) domain for Service Layer, as a business boundary of the Internet of Things platform. Within each SP domain, there is one Infrastructure node, and amount of MNs, ASNs and ADNs.

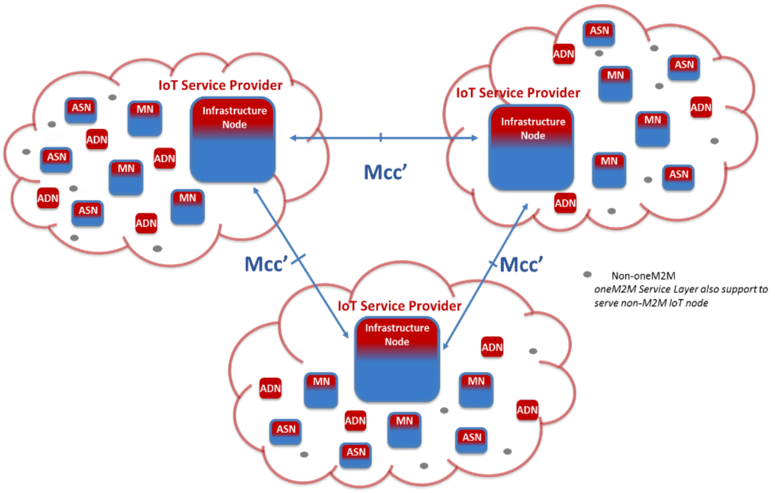
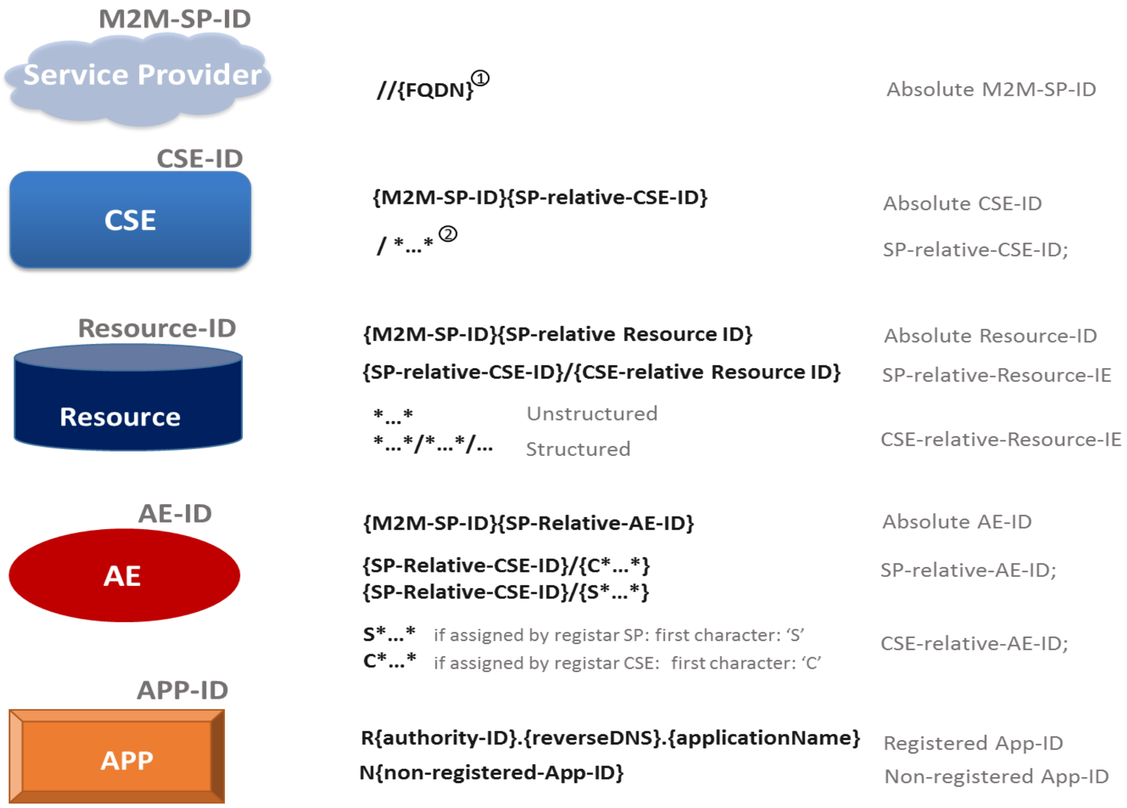


Figure 18: Inter-IoT Service Provider Communication

5.2. oneM2M identifiers

Each oneM2M CSE, as the service layer middleware is identified with a unique CSE-ID. Besides CSE, oneM2M also defines identifiers for an AE, a resource, a Node, as well as a SP to ensure the interoperability within an SP domain and between different SP domains. The format can be either absolute or relative. The oneM2M identifiers can be locally or globally unique depending on the context where they are allocated and the chosen format.


M2M-SP-ID
Service Provider
//{FQDN}①
Absolute M2M-SP-ID
CSE-ID
CSE
{M2M-SP-ID}{SP-relative-CSE-ID}
/ *…*②
Absolute CSE-ID
SP-relative-CSE-ID;
Resource-ID
Resource
{M2M-SP-ID}{SP-relative Resource ID}
{SP-relative-CSE-ID}/{CSE-relative Resource ID}
…
…/*…*/…
Unstructured
Structured
Absolute Resource-ID
SP-relative-Resource-IE
CSE-relative-Resource-IE
AE-ID
AE
{M2M-SP-ID}{SP-Relative-AE-ID}
{SP-Relative-CSE-ID}/{C*…*}
{SP-Relative-CSE-ID}/{S*…*}
S*…* if assigned by registar SP: first character: 'S'
C*…* if assigned by registar CSE: first character: 'C'
Absolute AE-ID
SP-relative-AE-ID;
CSE-relative-AE-ID;
APP-ID
APP
R{authority-ID}.{reverseDNS}.{applicationName}
N{non-registered-App-ID}
Registered App-ID
Non-registered App-ID

① FQDN format s defined in the IETF RFC 1035 [i.7]

② *…* represent any of the unreserved characters defined in the clause 2.3 of the IETF RFC 3986

Figure 19: oneM2M Identifiers Configuration

5.3. oneM2M resources

oneM2M is a resource orientated solution. All entities in the oneM2M service layer system, such as AEs, CSEs, data, etc. are represented as **Resources**.

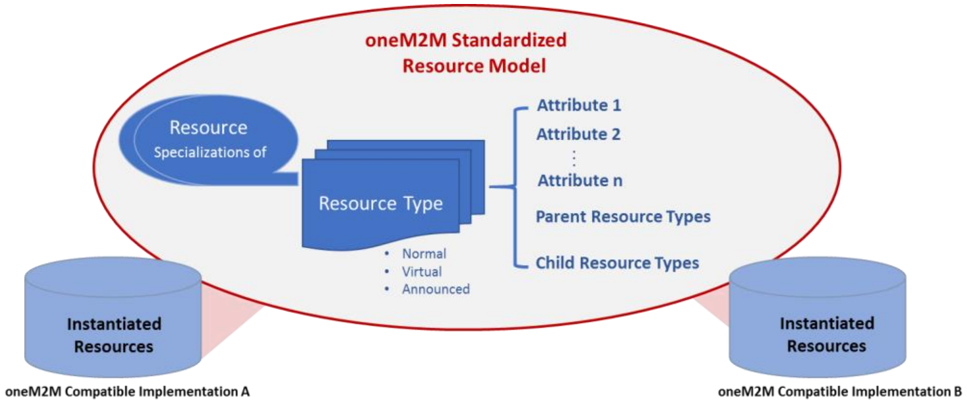


Figure 20: oneM2M Resource Models

oneM2M specifies the resource models for internet of thing service layer developer and adopter to implement an oneM2M compatible system. **Resource Types** are standardized, as well as their attributes, potential child resources and parent resources. oneM2M classifies Resource Types into ordinary and announced Resource Types. A resource of announced Resource Type is a resource at a remote CSE that is linked to the original resource that has been announced to facilitate the discovery of a resource at the remote CSE.

Table 1 lists 39 ordinary Resource Types standardized in oneM2M Release 2. There are other 13 announced Resource Types, of which each will have an addition of suffix "Annc" to the original Resource Type to indicate its associated announced resource type. For example, <*AEAnnc*> is the announced variant of <AE>. The announced resource maintains some of the characteristics of the original resource. In Table 1 Resource Type has an announced variant is written in **bold**.

Note The short name in the third column is used during transmission the Resource Type to reduce payload over telecommunication interface.*

*Note** Not all of the resource are required for a oneM2M compatible system, similarly not all specified attributes are mandatory for a Resource Type. It depends on the nodes type, functionalities.*

For a oneM2M CSE implementation, a resource of type <CSEbase> is the root of all the other resources that are hosted by the CSE. Resources in oneM2M system are linked by the means of 'linking' attributes, or the parent-child relationship. All resources can be addressed by Non-Hierarchical method or Hierarchical method. Figure 19 give an example of the relations between the relations and summarized the linked resources and the methods.

Table 1 oneM2M Resource Types

Resource Type	Funciton Hightlights	Short Name
accessControlPolicy	Stores a representation of privileges	
AE	Stores informaiton about the AE entity	
container	Shares data instances among entities	
contentInstance	Represents a data instance in the <container> resource	
flexContainer	A Template to define specialized version of containers	
CSEBase	The structure root for all the resources	
delivery	Forwards request from CSE to CSE	
eventConfig	Defines events that trigger statistics collection	
exeInstance	Contains all executions instances of the same management command	
fanOutPoint(V)	Virtual resource containing target for group request	
group	Stores informaiton about the resources of the same type	
latest	virtual resource points to moste recently created <contentInstance>	
locationPolicy	Includes information to obtain and manange geographical location	
mgmtCmd	Management command resource represent a method	
mgmtObj	Management object rsources represents management functions	
m2mServiceSubscription	Data pertaining to the M2M service subscription	
node	Represents specific Node Information	
notificationTargetMgmtPolicyRef	Represent a list of notification targets and the deletion policy	
notificationTargetPolicy	Represent a notification target deletion policy	
notificationTargetSelfReference(V)	Virtual resource used to remove the notification target	
oldest(V)	Virtual resource that points to the first created <contentInstance>	
pollingChannel	Represent a channel that can be used for a request-unreachable entity	
pollingChannelURI(V)	Virtual resource used to perform service layer long polling	
policyDeletionRules	Represent a set of rules	
remoteCSE	Represent a remote CSE	
request	Express/access context of an issued request	
schedule	Contains scheduling information for delivery of message	
serviceSubscribedNode	Node information	
statsCollect	Defines triggers for the IN-CSE to collect statistics for application	
statsConfig	Stores configuration of statistics for applications	
subscription	Respresent the subscription information related to a resource	
serviceSubscribedAppRule	A rule that defines allowed App and AE combinations that are acceptable for registering an AE on a Registar CSE	
sematicDescriptor	Semantic description pretaining to a resource and sub-resources	
semanticFanOutPoint	Virtural Resource as target for semantic discovery	
trafficPattern	Represent the communication& mobility pattern of a field node	
dynamicAuthorizationConsultation	Represent consultation informaiton for a CSE when performing consultation-based dynamic authorization	
timeSeries	Stores and shares time series data instances among entities	
timeSeriesInstance	Respresent a time series data instance in the <*timeSeries*>	

Among the Resource Types listed in Table 1, some Resource Type are defined like templates that can have different **Specializations** to provide specific functions.

<*mgmtObj*> is one of such Resource Type, it is specialized by designating an enumerated value of the *mgmDefinition* attribute. <*mgmtObj*> has 19 well-defined specialized Resource Types so far, including [*cmdhPolicy*], [*cmdhDefEcValue*], [*cmdhNetworkAccessRules*], [*areaNwkInfo*], [*deviceInfo*], [software], [firmware], [battery] etc. These specialization of <*mgmtObj*> provide device management, communication management and delivery handling for IoT device connected by wide or local area networks. <*mgmObj*> can also be specialized as 1-1 mapping to LwM2M objects for the interworking with LwM2M.

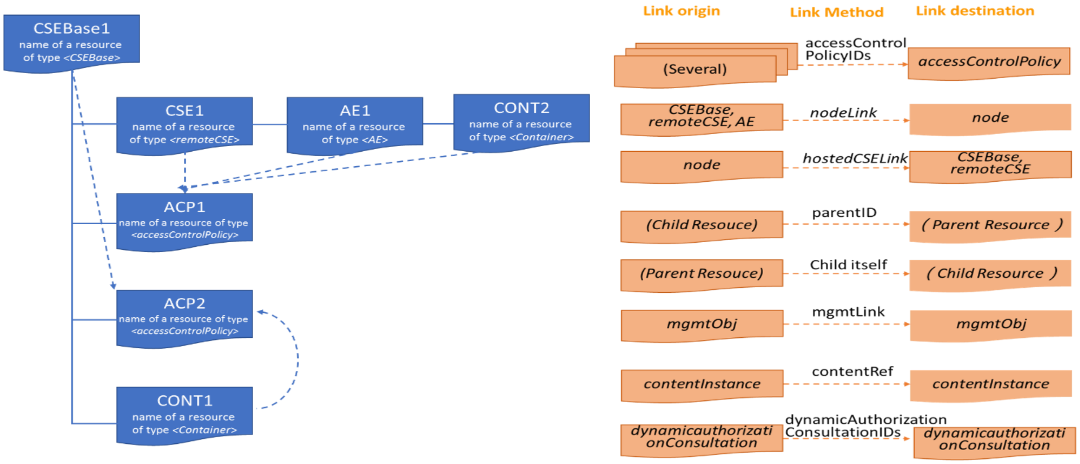
<*flexContainer*> is another example, which now has three groups of specializations for the use of interworking with other non-oneM2M systems.

I. Ontology-based Interworking: interworking with many types of non-oneM2M area networks and the devices via mapping the semantics to oneM2M Base Ontology.

II. Interworking with AllJoyn.

III. Interworking with Home Appliance Information Model.

oneM2M has a well-defined resources and resource architectures. It is especially worth to note that among the ordinary Resource Type defined by oneM2M, *<Container>, <contentInstance>, <flexContainer>, <timeSeries>, <timeSeriesInstance>* are termed as "Content Sharing Resources", they are used to carry the application data, and can be used to share data between CSEs, and oneM2M based service providers.


CSEBase1
name of a resource of type <CSEBase>
CSE1
name of a resource of type <remoteCSE>
AE1
name of a resource of type <AE>
CONT2
name of a resource of type <Container>
ACP1
name of a resource of type <accessControlPolicy>
ACP2
name of a resource of type <accessControlPolicy>
CONT1
name of a resource of type <Container>

Link origin	Link Method	Link destination
(Several)	accessControl PolicyIDs	*accessControlPolicy*
CSEBase, remoteCSE, AE	*nodeLink*	*node*
node	*hostedCSELink*	*CSEBase, remoteCSE*
(Child Resouce)	parentID	*(Parent Resource)*
(Parent Resouce)	Child itself	*(Child Resource)*
mgmtObj	mgmtLink	*mgmtObj*
contentInstance	contentRef	*contentInstance*
dynamicauthorizationConsultation	dynamicAuthorization ConsultationIDs	*dynamicauthorizationConsultation*

Figure 21: oneM2M Resources Addressing

5.4. oneM2M primitives

oneM2M entities communicate with each other via pairs of Request and Responses. Requests and Responses are exchanged between oneM2M originator and receiver as messages called “Primitives”. Standardized resources can be manipulated via **C**reate, **R**etrieve, **U**pdate, **D**elete and **N**otification operation of request Primitives.

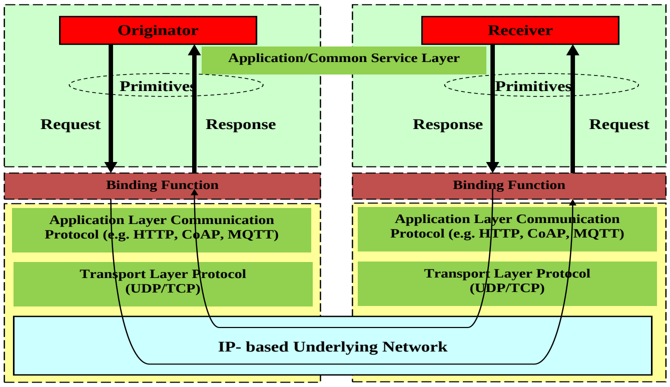

Originator
Receiver
Application/Common Service Layer
Primitives
Primitives
Request
Response
Response
Request
Binding Function
Binding Function
Application Layer Communication Protocol (e.g. HTTP, CoAP, MQTT)
Application Layer Communication Protocol (e.g. HTTP, CoAP, MQTT)
Transport Layer Protocol (UDP/TCP)
Transport Layer Protocol (UDP/TCP)
IP- based Underlying Network

Figure 22: oneM2M Primitives

There are different response types were defined, which indicates what type of response shall be sent to the issued Request Primitives and when the response shall be sent to the originator.

- nonBlockingRequestSynch: The Receiver CSE responds the Originator with an Acknowledgement confirming that the Receiver CSE will further process the Request. The Receiver CSE includes in the response to an accepted request a reference that can be used to access the status of the request and the result of the requested operation at a later time.
- nonBlockingRequestAsynch: The Receiver CSE responds the Originator with an Acknowledgement confirming that the Receiver CSE will further process the Request. The result of the requested operation needs to be sent as notification(s) to the notification target(s).
- blockingRequest: The Receiver CSE responds with the result of the requested operation after completion of the requested operation

Primitives have a number of mandatory and optional parameters. The primitive parameters’ presences depending on the create, request, update, delete or notification operations which was specified in [15] .

Each primitive is getting serialized in order to send it from one entity to the next. By oneM2M release 2, XML and JSON serialization are defined. And each primitive serialization is getting transported by mapping individual primitive parameters to specific element of the underlying transport binding (HTTP, CoAP, MQTT, WebSocket).

5.5. Common service functions

The services provided by oneM2M CSE are a set of common service functions (CSFs) that M2M applications across different industry segments commonly need. Figure 23 [16] shows examples of the services and a high-level description of oneM2M services are following. These functions are exposed to Applications via IT ‑ friendly APIs via the Mca reference point and to other CSE via Mcc reference points. These services can be extended and supplemented with oneM2M evolving.

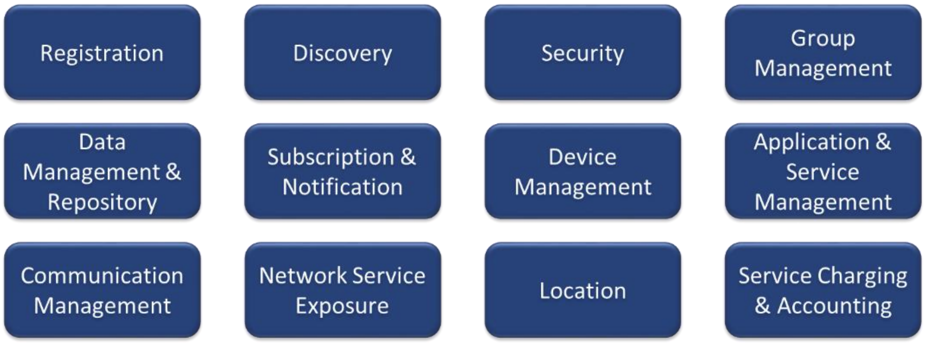


Figure 23 oneM2M Common Functions

- **Registration**

The Registration CSF processes a request from an AE or another CSE to register with a Registrar CSE in order to allow the registered entities to use the services offered by the Registrar CSE. Following a successful registration of an AE to a CSE, the AE is able to access, assuming access privilege is granted, the resources in all the CSEs that are potential targets of request from the Registrar CSE.

- **Discovery**

The Discovery CSF searches information about applications and services as contained in attributes and resources. The result of a discovery request from an Originator depends upon the filter criteria and is subject to access control policy allowed by M2M Service Subscription. An Originator could be an AE or another CSE. The scope of the search could be within one CSE, or in more than one CSE. The discovery results are returned back to the Originator.

- **Semantic Discovery**

The Semantics CSF enables applications to manage semantic information and provides functionalities based on this information. Thus, the Semantics Function brings value-added features related to the meaning of data and resources. The Semantics Function is based on semantic descriptions and supports features such as: annotation, resource filtering and discovery, querying, validation, mash-up, reasoning, analytics, etc. The Semantics Function also provides input for Access Control applied to semantic content and is responsible for the management of ontologies.

- **Group Management**

The Group Management (GMG) CSF is responsible for handling group related requests. The request is sent to manage a group and its membership as well as for the bulk operations supported by the group. When adding or removing members to/from a group, it is necessary to validate whether the group member complies with the purpose of the group. Bulk operations include read, write, subscribe, notify, device management, etc. Whenever a request or a subscription is made via the group, the group is responsible for aggregating its responses and notifications. The members of a group can have the same role with regards to access control policy control towards a resource. In this case, access control is facilitated by grouping. When the Underlying Network provides broadcasting and multicasting capability, the GMG CSF is able to utilize such capability.

- **Device Management and External Management operation**

The Device Management (DMG) CSF provides management of device capabilities on Middle Nodes (MNs) (e.g. M2M Gateways), Application Service Nodes (ASNs) and Application Dedicated Nodes (ADNs) (e.g. M2M Devices), as well as devices that reside within an M2M Area Network. Application Entities (AE) can manage the device capabilities on those Nodes by using the services provided by the DMG CSF alleviating the need for the AE to have knowledge of the technology specific protocols or data models. While the AE does not require an understanding of the technology specific protocols or data models, this information is provided to the AE so that an AE can utilize this information for administrative purposes (e.g. diagnostics, troubleshooting).

- **Location Management**

The Location (LOC) CSF allows AEs to obtain geographical location information of Nodes (e.g. ASN, MN) for location-based services. Such location information requests can be from an AE residing on either a local Node or a remote Node.

- **Subscription and Notification**

The Subscription and Notification (SUB) CSF provides notifications pertaining to a subscription that tracks event changes on a resource (e.g. deletion of a resource). A subscription to a resource is initiated by an AE or a CSE, and is granted by the Hosting CSE subject to access control policies. During an active resource subscription, the Hosting CSE sends a notification regarding a notification event to the addresses where the resource subscribers want to receive it.

- **Polling Channel Management**

An AE or a CSE that is request unreachable cannot receive a request from other entities directly. Instead this AE/CSE can retrieve requests that others sent to this AE/CSE once it created <pollingChannel> resource on a request reachable CSE. The request-unreachable entity polls any type of request(s) for itself from the <pollingChannel> Hosting CSE. For example, an AE can retrieve notifications by long polling on the channel when it cannot receive notifications asynchronously from a subscription Hosting CSE.

- **Service Charging and Accounting**

The Service Charging and Accounting (SCA) CSF provides charging functions for the Service Layer. It supports different charging models which also include online real-time credit control. The SCA CSF manages service layer charging policies and configuration capturing service layer chargeable events, generating charging records and charging information. The SCA CSF can interact with the charging System in the Underlying Network also. The SCA CSF in the IN-CSE handles the charging information.

- **Resource Announcement**

A resource may be announced from its Hosting CSE to one or more announcement target CSEs to inform the announcement target CSE(s) of the existence of the original resource. The announced resource also may be de-announced from the announcement target CSE(s). A limited set of attributes of original resource may be announced or de-announced in the resource announcement or de-announcement procedure.

- **Communication Management**

The communication management and delivery handling (CMDH) CSF provides communications with other CSEs, AEs and NSEs. It decides at what time to use which communication connection for delivering communications and, when needed and allowed, to buffer communications requests so that they be forwarded at a later time. This processing in the CMDH CSF is carried out per the provisioned CMDH policies and delivery handling parameters that can be specific to each request for communication. For communication using the underlying network data transport services, the underlying network can support he equivalent delivery handling functionality. There are more details in 5.6 on how CMDH works.

- **Security**

The oneM2M security solution provide configurable security services through an API for upper security domains to leverage or enable the use of the exposed security features of other security domains when appropriate. It connects data ‘producers’ and data ‘customers’ in a secure manner. Security would be the most important aspects for Internet of Things taking off, oneM2M provides a range of mechanisms addressing these concerns which can be found in 5.8.

5.6. Communication management and delivery handling

To manage a communication link for moving the sensor data is a most complicate task for application developer. For most non-oneM2M IoT platform solutions, the communication link is simplified as an end-to-end pipe, and data is moved ‘directly’ from one end of the “Pipe” to the other end of the “Pipe”. Usually, the communication link is not used efficiently, and application developers have few methods to configure and manage how the data be delivered across the communication network. On another hand, the communication operators have no idea on what kind of data is being delivered -- all the data were treated in the same way.

Besides transmission efficiency, delivery of data across different ‘communication network’ is another challenge for IoT application service provider. It would take them great resource to build a system that can control the device in different networks.

oneM2M is capable of control when communication occurs, depending on factors such as time-sensitivity of communications and the economics of data transfer. With specified multi-hops service layer architecture, oneM2M supports to access resources hosted on remote service platform via resource of <*delivery*> Resource Type. The application data could be transferred across different communication connections at target times, and when needed and allowed, the data and communication requests can be buffered so that they can be forwarded later.

Figure 24 gives an example for delivering information from source CSE1 to target CSE3.

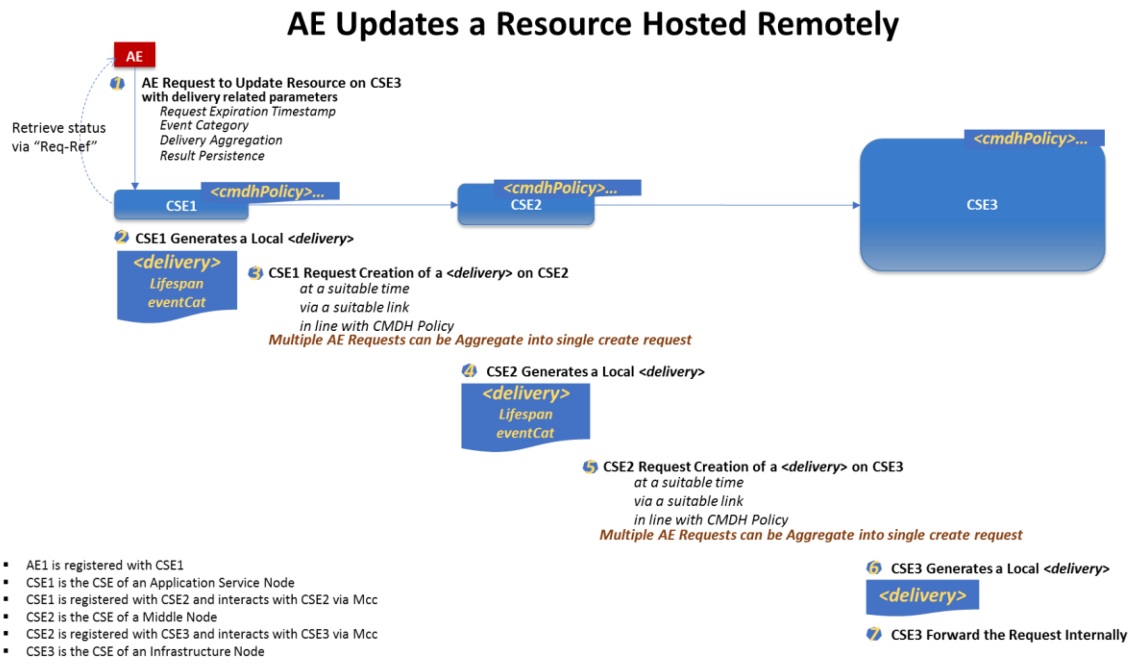

AE Updates a Resource Hosted Remotely
AE
1 AE Request to Update Resource on CSE3 with delivery related parameters
Request Expiration Timestamp
Event Category
Delivery Aggregation
Result Persistence
Retrieve status via “Req-Ref”
CSE1
<cmdhPolicy>...
CSE2
<cmdhPolicy>...
CSE3
<cmdhPolicy>...
2 CSE1 Generates a Local <delivery>
<delivery>
Lifespan
eventCat
3 CSE1 Request Creation of a <delivery> on CSE2
at a suitable time
via a suitable link
in line with CMDH Policy
Multiple AE Requests can be Aggregate into single create request
4 CSE2 Generates a Local <delivery>
<delivery>
Lifespan
eventCat
5 CSE2 Request Creation of a <delivery> on CSE3
at a suitable time
via a suitable link
in line with CMDH Policy
Multiple AE Requests can be Aggregate into single create request
AE1 is registered with CSE1
CSE1 is the CSE of an Application Service Node
CSE1 is registered with CSE2 and interacts with CSE2 via Mcc
CSE2 is the CSE of a Middle Node
CSE2 is registered with CSE3 and interacts with CSE3 via Mcc
CSE3 is the CSE of an Infrastructure Node
6 CSE3 Generates a Local <delivery>
<delivery>
7 CSE3 Forward the Request Internally

Figure 24 oneM2M Communication Management Example

The original Request is an UPDATE to a remote resource hosted on CSE3, which will include delivery related parameters including:

- ***Request Expiration Timestamp***: indicates how long the forwarding of the request can last
- ***Event Category***: indicates the event category that should be used by CMDH to handle this request
- ***Result Persistence***: indicates how long after the request expired, the local request context should still be available for retrieving status or result information.
- ***Delivery Aggregation***: would be set to ON indicating that <*delivery*> resource shall be used for forwarding the request.

AE shall get a confirmation from CSE1 when the original Request is accepted. With the provided reference (Req-Ref), AE can retrieve the status of the issued request later to find out if the request was already forwarded to CSE2 or if it is still waiting for being forwarded on CSE1.

In line with the delivery related parameters, CSE1 generates a local <*delivery*> resource and attempts to forward the content of it according to provisioned CMDH policies at a suitable time and via a suitable links to CSE2. In case there is demand to aggregate more than one original Requests into a single Request from AE, CSE1 could aggregate the Requests and send them together which could help to improve the transmission efficiency in some cases.

When CSE2 accepted the Request from CSE1, CSE2 creates a local <*delivery*> and attempts to forward it to CSE3.

When CSE3 accepted the Request to create a local <*delivery*>, CSE3 will determine that the target is itself, therefore to forward internally the original request contained in the data attribute of the <*delivery*> resource.

5.7. Service layer interworking with 3GPP network

oneM2M enables cellular network to be used as platform by 3rd parties using without having to deal with proprietary network vendor interfaces and facilitate expansion of cellular network usage into new vertical segments. The northbound APIs between the SCEF and the SCS/AS over T8 interface defined by 3GPP can be used by the oneM2M service layer that IoT applications can benefit from the relevant 3GPP features. Figure 25 shows the architecture of oneM2M interworking with 3GPP network.

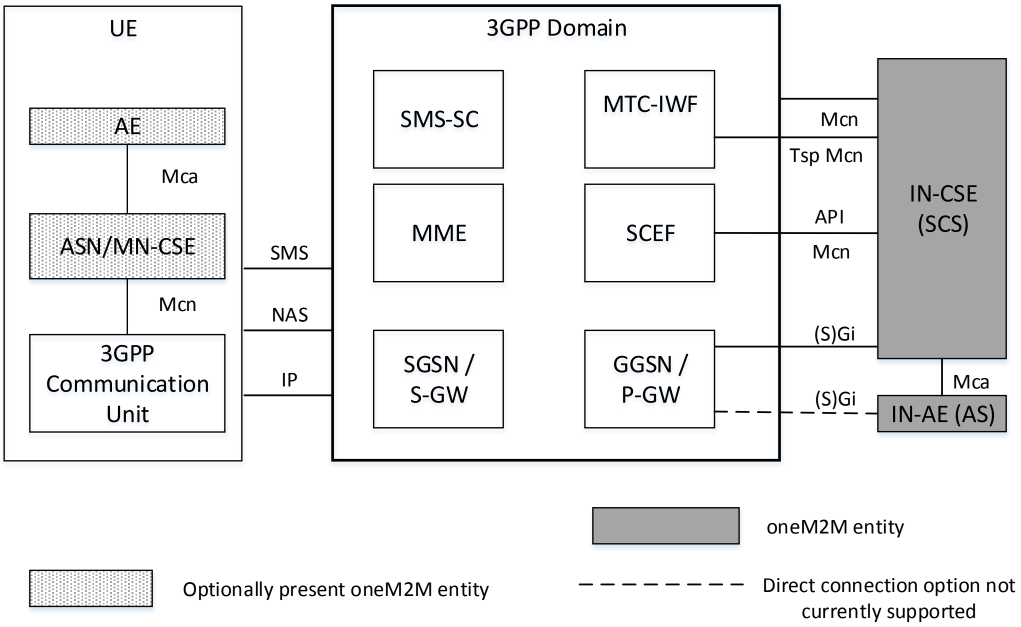


Figure 25: oneM2M Interfaces to the 3GPP Network [17]

oneM2M has supported below features exposed by SCEF of 3GPP.

- **Cellular IoT non-IP data delivery(NIDD)**

The 3GPP SCEF Non-IP Data Delivery (NIDD) functionality supports an API to allow the exchange of Non-IP data between an IN-CSE and an MN-CSE, ADN-AE, or ASN-CSE hosted on a UE. Via this SCEF NIDD API, an IN-CSE may exchange oneM2M request and response primitives with an MN-CSE, ADN-AE, or ASN-CSE hosted on a UE.

- **UE context information storage**

The UE context information that is stored in the different nodes for Machine Type Communications (MTC) device trigger procedure and NIDD procedures.

- **High latency communications**

The 3GPP SCEF High latency communication functionality supports an API to allow the IN-CSE to subscribe once and then get notification only when there has been some data delivery failure followed by the ADN/ASN becoming reachable.

An important use case for this functionality is the IN-CSE that wants to communicate with an ADN/ASN that sleeps for a long time. If downlink packets from the IN-CSE are not delivered, the IN-CSE recognizes that the ADN/ASN is not available by lack of response within a reasonable time from the ADN/ASN, and then await notification from the SCEF of ADN/ASN reachability.

- **Monitoring events**
  - ***UE Reachability monitoring***: The 3GPP SCEF functionality supports APIs for monitoring specific events such as UE Reachability status. This allows M2M Servers to request to receive reports when a device becomes reachable for receiving either SMS or downlink data.
  - ***UE Availability after DDN Failure***: The 3GPP SCEF functionality supports APIs for monitoring of events such as UE Availability after Downlink Data Notification (DDN) Failure. When communicating with UEs which sleep for a long time, if downlink packets are not delivered, the Underlying Network recognizes that the UE is not available by a lack of a response within a reasonable time.
  - ***UE Communication Failure***: The IN-CSE enables communications between large numbers of IN-AEs and devices in general and 3GPP UEs in the context of 3GPP Interworking. Informing the IN-CSE that devices have suffered communication failures in the Underlying Network helps optimize communications. For example, the IN-CSE may stop attempting to communicate with the device if it is aware of repeated the communication failures. The SCEF enables Communication Failure monitoring at the IN-CSE.
  - ***UE Loss of Connectivity***: The IN-CSE communicates with large numbers of devices, many of which are reachable for short periods of time. The IN-CSE wants to be informed when devices are not reachable to the Underlying Network, in order to better manage its communications. For example, IN-AEs which normally communicate with the device might not attempt communications if neither signalling or user plane communication are available.
  - ***Detecting Change of IMSI-IMEI(SV) Association***: The 3GPP SCEF Event Monitoring functionality supports an API that allows the IN-CSE to be informed when the SIM card of one physical device is placed in another physical device. This condition is detected by the underlying 3GPP network when the association between the International Mobile Subscriber Identity (IMSI) and International Mobile Equipment Identity (IMEI/IMEISV) changes.
  - ***Scheduling communication based on Roaming Indications***: The 3GPP SCEF Monitoring functionality supports an API to allow an IN-CSE to be informed when the roaming status of a UE in the underlying 3GPP network changes.
  - ***Location Reporting***: The 3GPP location monitoring feature supports reporting the current location of a UE as well as the last known location of UE, and provides location monitoring capability for an ASN/MN-CSE or ADN-AE hosted on a 3GPP UE.

- **3GPP Based Device triggering**

An IN-CSE may initiate a Device Trigger Request to an ASN/MN-CSE or ADN-AE to have it establish a connection to the IN-CSE, enrol to an M2M Enrolment Function (MEF), register to the IN-CSE, update its PoA, or perform a CRUD operation on a specified resource. The IN-CSE may initiate the Device Trigger Request itself or it may be initiated by a request that the IN-CSE receives from an AE.

- **Configuration of Traffic Patterns**

oneM2M uses the 3GPP MTC feature for configuration of device communication patterns to configure Node Traffic Patterns in the Underlying Network. To that purpose the IN-CSE translates the oneM2M Node Traffic Pattern (TP) into a 3GPP Device Communication Pattern.

- **Group message delivery**

For the oneM2M system, group message delivery allows the IN-AE/CSE to manage a group and its membership as well as to perform fanout operations to group member resources. When the same content is sent to the members of a group that are located in a particular geographical area, 3GPP provides MBMS capabilities that may be used to efficiently distribute the message to the group members using multicasting.

- **Informing about Potential Network Issues**

The 3GPP SCEF Network Status Monitoring functionality supports an API to allow an IN-CSE to be informed when there are network congestion issues in a particular geographical area in the underlying 3GPP network.

- **Setting up an AS session with required QoS procedure**

The 3GPP SCEF Setting up an AS session with required QoS functionality supports an API to allow the IN-CSE to request the network to provide QoS for the AS session based on the application and service requirements with the help of a QoS reference parameter which refers to pre-defined QoS information.

- **Background Data Transfer**

The purpose of this feature is to provide a means for the oneM2M System to inform the underlying network of parameters that can be used for optimizing the background data traffic over the underlying network for a set of Field Domain Nodes (UEs). Such parameters may include the expected number of UEs in the set and amount of data to be transferred a desired/preferred time window for the data transfer to these UEs, and network area information. In response, the underlying network may inform the oneM2M system about policies that may be used to meet the given background data transfer request.

- **Change the chargeable party at session set-up or during the session procedure**

The IN-CSE request the SCEF to start or stop sponsoring a data session for an ASN/AND that is served by the 3rd party service provider (AS session), i.e. to realize that either the 3rd party service provider is charged for the traffic (start) or not (stop). The IN-CSE may request to be set as the chargeable party, i.e. sponsoring the traffic, either at AS session set-up or to change it during an ongoing AS session.

- **Network Parameter Configuration**

The 3GPP SCEF functionality supports an API for Network Parameter Configuration which may be used by the IN-CSE to suggest to the 3GPP Mobile Network specific configuration parameters. The procedure may be used by the IN-CSE to influence certain aspects of UE/network behavior such as the UE's PSM and extended idle mode DRX. For this purpose, parameter values may be suggested for Maximum Latency and Maximum Response Time for a UE. The Mobile Core Network may choose to accept, reject or modify (via the SCEF) the suggested configuration parameter value.

- **Node Schedule Management**

In the context of 3GPP connectivity technologies, the network reachability and UE reachability are both indications that the UE becomes reachable for receiving either an SMS or downlink data. The SCEF supports the capability to notify the IN-CSE of the network reachable status or the UE reachable status. The IN-CSE shall maintain a <*schedule*> resource of a UE and if the *networkCoordinated* attribute of the <*schedule*> is set to True, then the IN-CSE shall coordinate the schedule based on the UE’s reachability. For example, the IN-CSE shall support synchronizing the start time of the *scheduleElement* attribute to be the same as the start time of the targeted UE idle status which the IN-CSE receives from the Underlying 3GPP Network.

5.8. **Security**

The ability to remotely access IoT/M2M devices is a blessing, but it comes with a curse - the threat of malicious parties remotely compromising those devices or remotely compromising the data exchanged across IoT deployments. It is assumed that the readers of this document are aware of the security and privacy concerns for their systems.

Security and privacy concerns are the #1 barrier for taking off of the Internet of Things, and oneM2M provides a range of mechanisms addressing these concerns.

5.8.1. Challenges of IoT security

A universally-applicable IoT system like oneM2M faces a variety of challenges in being adaptable to the wide range of IoT devices, business needs and personal privacy decisions.

IoT devices will range from cheap "convenience" devices (which require only low-complexity security due to the limited ability to impact safety)to expensive, safety-critical sensors and actuators which require more complex security mechanisms.

Each IoT vertical market has its unique business needs, and satisfying this range of requirements is a challenge in itself. However, a single vertical supports a range of businesses, and even an individual business can have a range of business needs. For example,

- Some businesses adopting IoT solutions might prefer using public key certificates, while others prefer configuring shared keys.
- Each business can have different policies for authorizing access to create, retrieve, update and deletion of resources. A business can have a range of access policies depending on whether a request is reading a sensors measurement, reading an actuator's current settings, controlling an actuator, reading configuration or updating configuration.
- Some use cases have strict requirements for end-to-end security. Other use cases extract value by sharing the data widely, in which case end-to-end security is an obstacle.

A universally-applicable IoT system security framework needs to provide a high degree of flexibility to adapt to all kind of business needs.

A third challenge is the range of people's preferences regarding sharing of their personal data (privacy) with Application Service Providers. A person's preference can vary based on the Application Service Providers policies. There will be an increasing volume of personal data, an increasing number of Application Service Providers and an increasing variety of privacy policies. Reviewing each privacy policy will quickly become untenable. A flexible, scalable system for managing privacy policies and authorizing sharing of personal data can be a valuable component of a universally-applicable IoT system security framework.

5.8.2. Overall approach of oneM2M on security

The core purpose of oneM2M security can be summarized in a nutshell as follows: When a device is operational, messages are exchanged over mutually-authenticated secure channels between oneM2M entities. The mutual authentication enables the sending and receiving oneM2M entities to verify each other's identity (CSE-IE or AE-ID) and any possibly assigned access roles. The secure channel provides encryption, integrity and replay protection for the messages. CSEs use the identities and roles when deciding whether to permit or deny request received over the secure channel.

Note that oneM2M allows exchanging messages without secure channel, but provides no details about how oneM2M identities are associated with such messages. the present description assumes that messages are exchanged over a secure channel.

oneM2M provides many other security functions, but they are all the security functions which are built around supporting this core purpose.

Identification and Authentication. When operational, mutual authentication is based on one of three mechanisms: provisioned symmetric keys (also known as shared keys) or public key certificates or an M2M Authentication Function (MAF) which is a centralized function facilitating authentication. Once authenticated, identity and one or more assigned roles can then be associated with the oneM2M entity.

Remote Provisioning Each mutual authentication mechanism relies on provisioning the oneM2M entities with credentials, identifiers, ciphersuites and other security parameters. oneM2M has defined the M2M Enrolment Function (MEF) with mechanisms for remotely provisioning these security parameters.

Communication security: Mutually authenticated channels can be established "point-to-point" between adjacent oneM2M entities or "end-to-end" between oneM2M entities communicating across multiple intermediate nodes.

Authorization and access control: oneM2M supports fine-grained access control, with access decisions made either on the resource host (governed by configurable policies) or by a trusted authorization server. The access decisions are evaluated based on the authenticated identity and/or roles of the originator.

Wherever possible, oneM2M uses open, standardized security protocols and cryptographic algorithms.

5.8.3. Security frameworks specified by oneM2M

The oneM2M security frameworks are specified in TS-0003 [18]. The remote provisioning of security parameters to field devices by means of device management mechanisms is defined in TS-0022 [19]. The protocol used for communication with the M2M Authorization Function (MAF) and M2M Enrolment Function (MEF) is specified in TS-0032 [20]. TR-0008 [21] provides an analysis of security threats.

oneM2M has defined the following security services:

- Secure enrolment services (Remote Security Provisioning Frameworks, RSPF)
  - Credentials Provisioning/Security Configuration of the M2M System
- Secure communications services (Security Association Establishment Frameworks, SAEF)
  - Methods for Securing Information
- Point-to-point (TLS / DTLS) and end-to-end solutions using JSON Web Encryption/Signature
- Access Control & Authorization services
  - Requester Authentication
  - Information Access Authorization
  - Static (ACL-based) and Dynamic (token-based) solutions
- Privacy Policy Management

In oneM2M systems four levels of operational security can be differentiated:

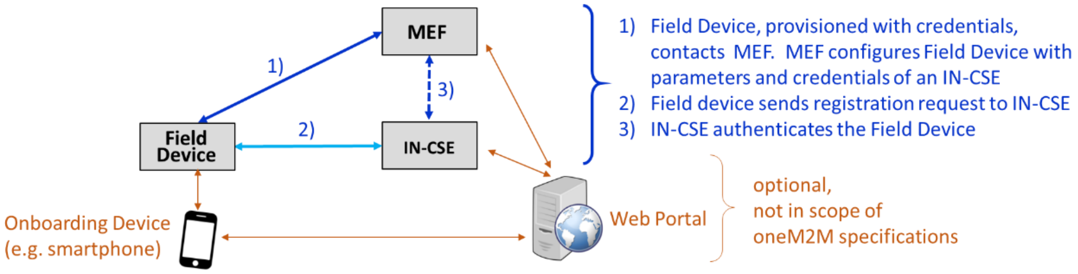
- No use of onM2M-specific security: This could be suitable for M2M devices protected from attackers by mechanisms outside the scope of oneM2M, i.e. when communicating via trusted networks which employ network-specific security mechanisms, e.g. VPN or secured cellular radio transmission.
- Use of software-based only security: This is denoted as white-box cryptography (WBC) which allows to perform cryptographic operations without revealing any portion of confidential information such as e.g. cryptographic keys. Software-based security is still vulnerable to sufficiently motivated attackers. However, this security level is still acceptable when a potential compromise is not extremely critical.
- Use of Trusted Execution Environment (TEE): Software implementing security mechanisms runs in a TEE which relies on special hardware features of the main CPU of an M2M device. This provides a very good barrier against software based attacks. It is sufficient for remotely accessible, but not physically exposed M2M devices.
- Use of tamper resistant hardware embedded Secure Element (eSE): Use of eSE technology provides the highest possible level of security. Any critical information is available on tamper resistant hardware even if the (possibly unattended M2M device is physically exposed to attackers.

5.8.3.1. Secure enrolment services

Secure enrolment comprises a number of procedures which need to be executed to prepare both the M2M system and an M2M field device with parameters and credentials in order to allow the field device to perform initial registration with its target registrar CSE.

In the oneM2m architecture, enrolment services are facilitated by an M2M Enrolment Function (MEF).

Field devices can either be preconfigured to initially contact a MEF, or can be configured with the help of an on-boarding device (e.g. a smartphone) to contact a MEF. Communication protocols and procedures used by on-boarding devices are not standardized by oneM2M and left to implementation.


MEF
1)
3)
2)
Field Device
IN-CSE
Onboarding Device (e.g. smartphone)
Web Portal
optional, not in scope of oneM2M specifications

1) Field Device, provisioned with credentials, contacts MEF. MEF configures Field Device with parameters and credentials of an IN-CSE
2) Field device sends registration request to IN-CSE
3) IN-CSE authenticates the Field Device

Figure 26 M2M enrolment service

The communication interface between the field devices and the MEF is specified in TS-0032 [20]. The MEF performs remote provisioning of security parameters to field devices. oneM2M has defined three different Remote Security Provisioning Frameworks (RSPF) which are differentiated by the type of security scheme applied between MEF clients and the MEF. These are

- Symmetric key authenticated RSPF
- Certificate authenticated RSPF
- GBA-authenticated RSPF; in this case the MEF is the Bootstrapping Server Function (BSF) of the 3GPP Generic Bootstrapping Architecture (GBA).

The MEF furthermore triggers field devices to perform a variety of procedures e.g. to generate Certificate Signing Requests in order to reassign new certificates before old ones expire, and to configure field device configuration parameters by means of external Device Management procedures such as BBF TR-069, OMA-DM, and LwM2M. Procedures which can be triggered by the MEF are specified in oneM2M TS-0003 [18] and TS-0022 [19].

5.8.3.2. Operational security services

5.8.3.2.1. Overview

oneM2M has defined three types of operational security frameworks which tie together credential management, configuration parameters, establishment of a security session (by TLS/DTLS handshake) and protection of the exchanged messages (oneM2M primitives) or of individual data elements of a message:

- **Security Association Establishment Framework (SAEF):** these security mechanisms establish a security association between adjacent oneM2M entities, i.e. between registree AEs or CSEs and registrar CSEs. This includes mutual authentication, key agreement and secure communication by means of a TLS or DTLS handshake procedure.
- **End-to-End Security of Primitives (ESPrim):** these security frameworks provide end-to-end security of messages from the originator to the receiver across intermediate nodes (i.e. MN-CSEs). With ESPrim, oneM2M requests can be encrypted entirely and transported as payload of a Notify request to the receiver. Only the receiver has the credentials to decrypt the request and perform the requested action.
- **End-to-End Security of Data (ESData):** these security frameworks provide mechanisms to protect individual objects of a primitive, for instance the content attribute of a <contentInstance> resource instance, which may include critical personal information. This information is then carried end-to-end from the originator of the information to the consumer of the information. JSON Web Encryption/Signature or XML Encryption/Signature mechanisms can be used for ESData.

5.8.3.2.2. Security association establishment

oneM2M permits three types of Security Association Establishment frameworks:

Provisioned Symmetric Key (PSK) Security Association Establishment: A symmetric key is provisioned to each entity, along with the CSE-ID or AE-ID of the other entity with which the symmetric key is shared. The entities authenticate each other by verifying Message Integrity Codes in the Security Handshake which were generated using the symmetric key. After successful authentication, each entity associates the session with the CSE-ID or AE-ID provisioned with the symmetric key.

Certificate-Based Security Association Establishment: The entities are each provisioned with:

- a Private Signing Key that is known only to that entity;
- a Certificate containing the corresponding Public Verification Key and an identifier: either a globally unique hardware identifier or a CSE-ID or AE-ID; and
- (Optionally) a Certificate Chain from the entity's Certificate to a Root Certificate.

The entities are also configured with Root Certificates for validating other entity's certificates. The entities validate each other's Certificate before trusting the Public Verification Keys in the Certificate. Within the Security Handshake, the one entity ((D)TLS client) creates a digital signature of the session parameters using its private signing key and the other entity ((D)TLS server) verifies the digital signature using the client entity's public verification key. Then the roles are reversed: the serving entity creates a digital signature and the client entity verifies it. After successful authentication, each entity associates the session with the identifier in the other entity's certificate.

M2M Authentication Function (MAF)-based Security Association Establishment. This Security Association Establishment Framework uses mutual authentication of a field node and a M2M Authentication Function (MAF) and derive a M2M Secure Connection key for use with the registrar CSE of the field node, and the MAF delivers this key along and field node identifier to this registrar CSE (via separate mutually-authenticated communication). The entities then authenticate each other using the M2M Secure Connection key. After successful authentication, the field node associates the session with the CSE-ID provided to the MAF, while the registrar CSE associates the session with the identifier provided by the MAF. Both entities can use either symmetric key credentials or certificates for mutual authentication with the MAF.

Figure 27 shows an example where a security association SA1 between an ADN-AE and an MN-CSE is established using DTLS. This applies when e.g. CoAP binding and UDP/IP transport is used between these entities. A security association SA2 using TLS is established between the MN-CSE and an IN-CSE. This applies when e.g. HTTP binding and TCP/IP transport is used between MN-CSE and IN-CSE. SA1 is assumed to be established using either PSK or Certificate-based Security Association Establishment. In this example, SA2 is established using MAF-based Security Association Establishment (security associations between the MN-CSE and the MAF and the IN-CSE and the MAF are not shown in the figure for simplicity). Note that the information exchanged over SA1 becomes visible in the clear to the MN-CSE, i.e. the MN-CSE must be trusted, unless end-to-end security services are employed additionally.

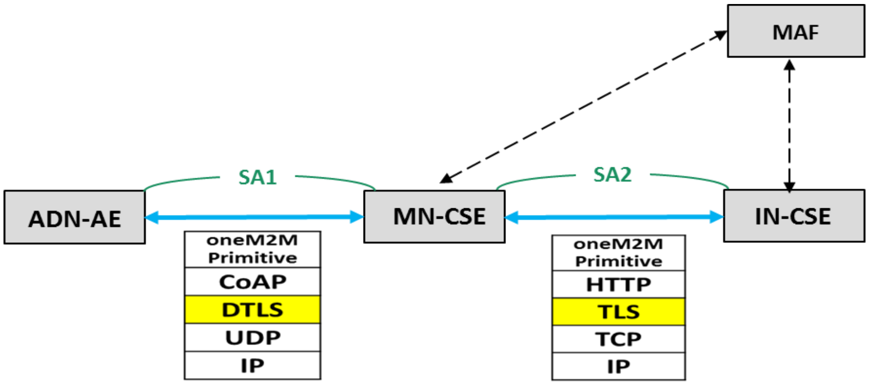


Figure 27 Security Association Establishment

5.8.3.2.3. End-to-End security of primitives

End-to-End Security of Primitives (ESPrim) is an interoperable framework for securing oneM2M primitives. When ESPrim is used, request and response primitives exchanged between an originating end entity and a receiving end entity. Intermediate IN-CSEs which forward the primitives do not need to be trusted. ESPrim provides mutual authentication, confidentiality and integrity protection between the end entities. JSON Web Encryption (JWE) as specified in IETF RFC 7516 is the format to be used for encryption of ESPrim objects using a symmetric key which can be established by remote provisioning (using a MEF), End-to-end Certificate-based Key Establishment (ESCertKE), or oneM2M authentication server (MAF).

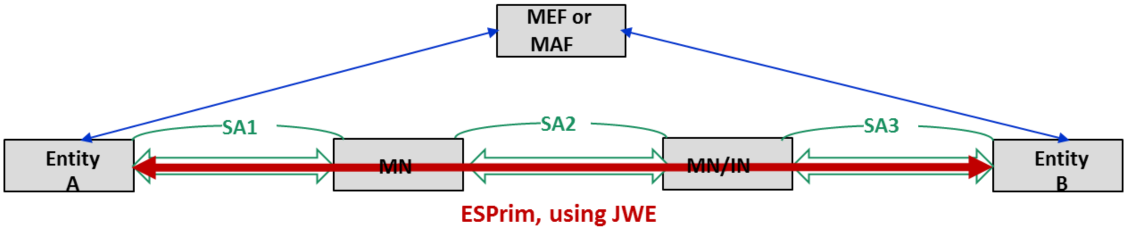


Figure 28 Illustration of end-to-end security of primitives (ESPrim)

5.8.3.2.4. End-to-End security of data

End-to-End Security of Data (ESData) is an interoperable framework for protecting a selected data portion of a primitive. The data to be protected is denoted as ESData Payload. Any transited CSEs do not need to be trusted with that data. The ESData framework is illustrated in Figure 29.

ESData payload could compose all or part of an attribute value (e.g. *content* attribute value of a <*contentInstance*> resource) or a primitive parameter (e.g. a signed, self-contained access token communicated in a request primitive to obtain dynamic authorization).

The protocol applied for ESData in oneM2M systems is JSON Web Encryption/Signature (JWE/JWS) or XML Encryption/Signature.

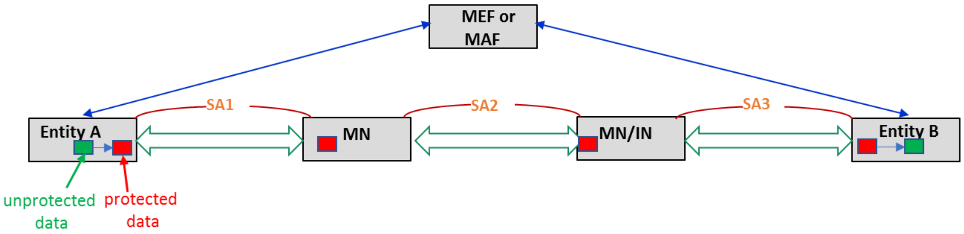


Figure 29 Illustration of end-to-end security of data (ESData)

5.8.3.3. Authorization

The M2M authorization procedures control access to resources and services hosted by CSEs. The authorization procedure requires that the originator of the resource access request message has been identified and authenticated to the receiver. oneM2M has defined two types of access control mechanisms: authorization by using static access control lists; and dynamic authorization.

5.8.3.3.1. Authorization using access control lists

On a computer file system, access to files of data is typically controlled by assigning a set of access control permissions to each file which control what type of user (e.g. individual user, user group, all users) can perform which operation (e.g. read, write, delete) on a specific data file. This file access information is typically included in the file header.

A similar, but more sophisticated approach, is employed in oneM2M systems. Each resource addressed in a request message has an associated *accessControlPolicyIDs* attribute (either included explicitly as an attribute of the resource addressed in the request message, implied from the parent of the resource, or set fixed by the system). The *accessControlPolicyIDs* attribute contains a list of identifiers of <*accessControlPolicy*> resources applicable to the resource addressed in the request message.

Figure 30 illustrates the relation between <*accessControlPolicy*> resource instances (ACP) and the instances of the protected resources, denoted Resource_1 to Resource_N. A “link” between a Resource instance and an ACP is created by an entry in the *accessControlPolicyIDs* attribute of the resource. The concept of using links allows to assign a specific ACP instance to multiple resource instances. Each ACP instance includes one or more “ACP Rule(s)”. Each such ACP Rule describes *who* (originator identified by AE-ID, CSE-ID or Role-ID) is allowed to perform *what* operation (Create, Retrieve, Update, Delete, Discover, Notify) under *which* specific context constraints on the resource to which the ACP applies. Context constraints can be defined as optional part of an ACP. These can demand that access is allowed only within specific time windows, from a specific geographical location of the originator, with a specific IP source address, for authenticated or unauthenticated users, and/or request object details such as a set of specific resource types.

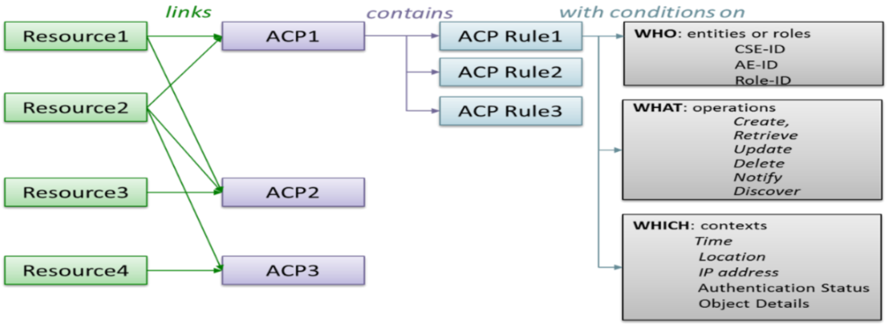


Figure 30 Illustration of Access Control Policies assigned to resources

If the parameters associated with a request message satisfy any of the rules contained in a ACP linked form the requested resource, then the access control system grants access and executes the request. If this requirement is not fulfilled, then the access request is rejected.

5.8.3.3.2. Dynamic authorization

Dynamic Authorization provides an interoperable framework for an Originator to be dynamically issued with temporary permissions providing the Originator with access to one or more resources on one or more CSEs. The access authorization is provided by a Dynamic Authorization System (DAS) Server.

Two variants of Dynamic Authorization may be supported by a oneM2M system:

- Direct Dynamic Authorization: Hosting CSE submits request to the DAS Server, with the Originator not communicating with the DAS Server.
- Indirect Dynamic Authorization: Originator submits request for authorization to the DAS Server using information received from a Hosting CSE. The DAS returns a token for the Originator to include with the resubmitted request. The token can include a cryptographically signed description of the associated authorization, or the Hosting CSE can retrieves this description from the DAS Server. This scheme is similar to the Open Authentication (OAuth) mechanism.

For both variants, the DAS Server has multiple options for authorizing a request. It can create temporary applicable ACPs, assign a specific Role to the Originator which allows access, or issue JSON Web Tokens (JWT). If the token includes the description of the associated authorization, then the JWT is signed using ESData.

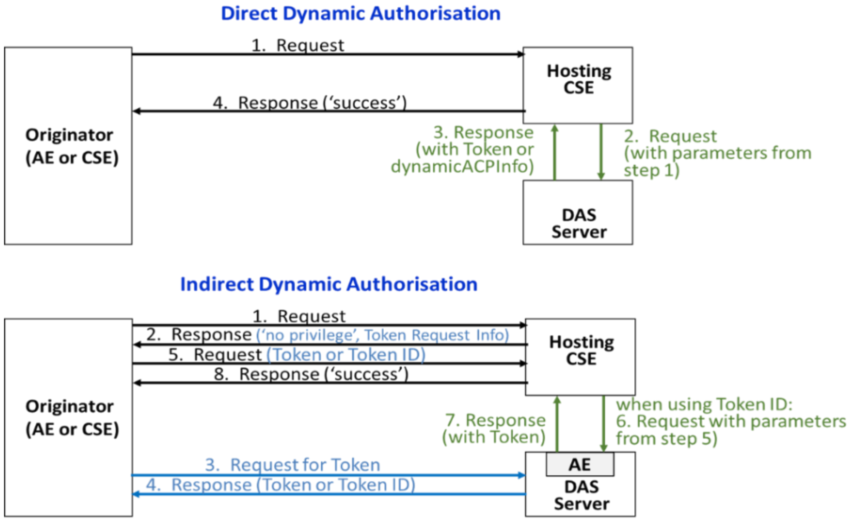

Direct Dynamic Authorisation
1. Request
Hosting CSE
4. Response ('success')
Originator (AE or CSE)
3. Response (with Token or dynamicACPInfo)
2. Request (with parameters from step 1)
DAS Server
Indirect Dynamic Authorisation
1. Request
2. Response ('no privilege', Token Request Info)
5. Request (Token or Token ID)
8. Response ('success')
Hosting CSE
Originator (AE or CSE)
7. Response (with Token)
when using Token ID:
6. Request with parameters from step 5)
3. Request for Token
4. Response (Token or Token ID)
AE
DAS Server

Figure 31 Dynamic Authorization

5.8.3.3.3. Privacy policy management

The Privacy Policy Manager (PPM) is a framework enabling alignment between Application Service Provider’s (ASP's) privacy policies and the service subscriber’s (i.e. user of M2M services) privacy preferences. The PPM manages access to a user's Personally Identifiable Information (PII) which is either stored on a CSE or accessible via a CSE. The PPM may be operated by an M2M Service Provider or another stakeholder acting as trusted third party.

The PPM obtains users' privacy preferences and ASPs' privacy policies in a machine-interpretable format. The PPM can determine the matches and mismatches between an ASP's privacy policies and a user's privacy preferences. The PPM can then indicate, to the user, which of ASP's privacy policies fall outside the user's privacy preferences. The user can then focus on those ASP's privacy policies which are of more interest to the user, thus simplifying the user's decision to accept or decline the ASP's privacy policies. If the user accepts the ASP's privacy policies, then the PPM will provide the ASP with authorization to access the user's personal information.

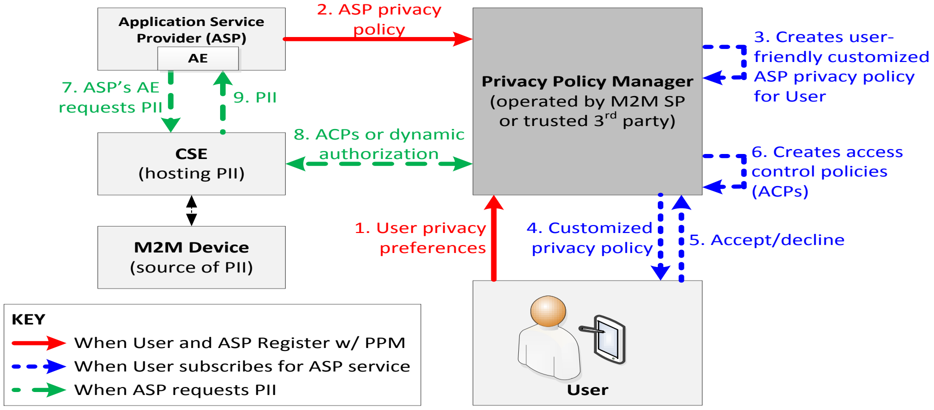

Application Service Provider (ASP)
AE
2. ASP privacy policy
Privacy Policy Manager (operated by M2M SP or trusted 3rd party)
3. Creates user-friendly customized ASP privacy policy for User
7. ASP’s AE requests PII
9. PII
8. ACPs or dynamic authorization
CSE (hosting PII)
6. Creates access control policies (ACPs)
M2M Device (source of PII)
1. User privacy preferences
4. Customized privacy policy
5. Accept/decline
User
KEY
When User and ASP Register w/ PPM
When User subscribes for ASP service
When ASP requests PII

Figure 32 Privacy Policy Management

5.9. Interworking

The current IoT-related standards and technologies are highly fragmented. The fragmentation can be seen across different verticals/applications domains where there is very little or no re-use of technologies. Given that multiple standards have always existed – many of them quite well established in their own areas, it is not realistic to expect the ecosystem to converge on a single standard.

Among the plenty of alliances with the intention of establishing standards conductive to the implementation of IoT, oneM2M stands out as a unifying standard for most of these initiatives. oneM2M can be seen as a “standard of standards”, it is involved in concerted efforts to bring in interoperability among architectural layers across IoT applications.

oneM2M release 1 and 2, as well as the on-going release 3 specifications are already addressing interoperability for many of the common existing industry standards and technologies[29], and specifying framework for non-specific technology related general interworking.

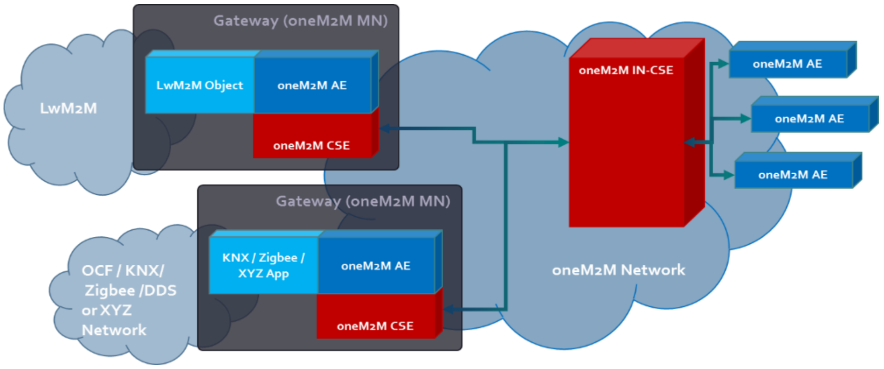


Figure 33 Interworking Framework Example with Other Technologies

Specific Technology Interworking：

- LwM2M Interworking: Providing transparent transport of encoded LwM2M application objects between Lwm2M endpoints and M2M applications. It is also specifying full mapping of LwM2M objects in LwM2M endpoints to semantically enabled resources than are utilized by M2M applications [22].
- AllJoyn Interworking: Specifying the oneM2M and AllJoyn interworking that enable AllJoyn Applications and oneM2M entities produce/consume services, describing AllJoyn services to oneM2M mapping structure and rules[23].
- OIC Interworking: Providing transparent transport of encoded OIC resources and commands in oneM2M resource types between OIC devices and M2M applications[24].
- OSGi Interworking: Ongoing [25]
- DDS Interworking: Ongoing
- Modbus interworking: Ongoing
- W3C Web of Thing interworking: ongoing

- Global Platform Interworking: ongoing

Non-specific Technology Interworking

- Home Appliances Information Model and Mapping: Describes the oneM2M defined information model for home appliances, including the description of a method on how it is mapped with other information models from external organizations which include AllJoyn, OIC, HGi Smart Home Device Template (SDT) and ECHONET [26].
- Ontology based general interworking: Provide interworking with only specified data models which can be flexibly be provided in form of a formally described ontology. This applies to those companies which want to publish their proprietary data model for interworking purposes but does not wish to reveal their proprietary technology (radio technology, communication protocol.) for data transmission [27].
- Proximal IoT Interworking: Enabling the exchange of information and the use of services irrespective of whether they are designed as oneM2M-defined entities or other non-oneM2M-defined proximal IoT technologies [28].

6. Communication service suite for IoT

6.1. Communication service suite architecture

In order to drive the cellular IoT market size and stimulate new business growth, China Mobile setup cloud side platform OneNET for IoT. While, there are so many kinds of IoT devices and they are diversified. For many IoT devices, resources are constrained. For cost considerations, IoT devices can only use limited resources, and their CPU, Flash and RAM resources are constrained. Getting the data transmitted to the cloud side was not a simple and easy thing. Developers usually find that they are faced with the following challenges:

- For IoT solution development and porting, multiple choices of chipsets and modules may lead to heavy workload and strict entry criteria.
- Different kinds of communication protocols and semantic systems may result in isolated IoT systems.
- MNOs and other solution providers may expect to make their IoT systems interoperable for value-added services and big data analysis.

To address the above challenges for IoT application developers, China Mobile Research Institute designed the Communication Service Suite (ComSS) to enable IoT devices easily connecting to OneNET. This ComSS defines a simple and easy to use APIs for upper layer applications, and specify network adapter interface to hide the difference of the underlying network. To drive large scale deployment of IoT devices, standardized LwM2M protocol[34] was selected as the interface to the cloud side platform. Since Nov. 2017, cellular IoT devices can get easily access to OneNET with the ComSS.

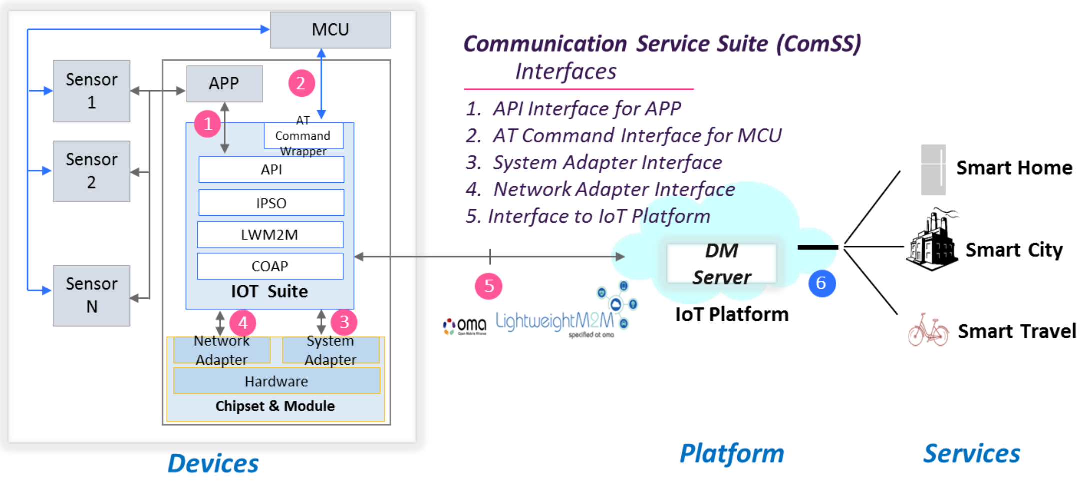


Figure 34 China Mobile Communication Service Suite Architecture

As showed in Figure 34, the communication service suite in device side defines interfaces, ①,②, ③,④,⑤.
① API interface to application, IPSO smart object are supported.
② AT command set to MCU when application run in MCU;
③ System adapter API, it hides the differences for different chip/module, including system memory operation, system time obtainment, and random number obtainment and so on;
④ Network adapter API, it hides the differences for the network, and it is used for physical network connection establishment, and network data transmission and reception;
⑤ The interface between device and IoT platform defined by OMA LwM2M;
Chipset & module vendors can provide the APIs to their customers on the device side. This communication suite implementation can be seamlessly ported to different chips and module, to help developers to develop their own application.

On the platform side, simple and easy to use APIs to IoT applications is also a must-have for applications, which would support the IoT service to display, analyse and use the data, which are showed in the figure as interfaces ⑤, ⑥.
⑤ the interfaces between device and IoT platform defined by OMA LwM2M.
⑥ the Restful API to AS.

The communication service suite in device is implemented as a LwM2M client, it encapsulates protocols including the UDP-based CoAP at the bottom layer, CoAP-based LwM2M at the middle layer, and profile specification used in LwM2M at the upper layer.

Based on the communication service suite architecture, applications can be located in different part. When application and module/chip are integrated in the same chip, it uses the APIs to communicate with the IoT Platform. In another scenario, when the application integrated in individual MCU, the application uses AT commands to communicate with communication service suite to talk to IoT platform.

On IoT device side, the functions provided by the Communication Suite are just like a simple Service Layer illustrated in oneM2M on the ASN side, which hides the diversity of the underlying-layer system and bottom-layer network and provide simple interfaces to applications. On IoT platform side, similar functions are also needed to hide the diversity of underlying system for IoT applications.

China mobile's experience showed that standardization of the interface sets and extension to the definition of the IPSO smart objects is necessary to drive the IoT device and AS development and promote the IoT industry. To have the IoT industry using unified invocation interface via standardization will simplify the application development and help the operator to run the values.

6.2. Common functions of communication service suite

There are three sets of functions provided by ComSS. The first part is interfaces provided to application, including API interfaces、AT commands, and the preloaded objects for devices; The Second part is interfaces between UE and IoT Platform; The last one is Restful APIs provided by IoT platform.

For the first part, APIs provides uniform interface for Init, Register, DeRegister, AddObject, Deleteobject, Notify, UpdateRegister and DeInit operations. They hide the details about how to communicate between the client and IoT platform. The functions are discribed as below:

- **Init**: initialize the service layer based on the inputed parameters such as the server address, port, APN name, APN password,etc.
- **Register**: register the device to the IoT platform with the objects.
- **DeRegister**: deregister the device to the IoT platform with the objects.
- **AddObject**: add object to the service layer with the object information.
- **DeleteObject**: delete object from the service layer.
- **Notify**: report the values for the object to IoT platform.
- **UpdateRegister**: update the registration information to the IoT platform such as lifetime and object information.
- **DeInit**: destroy the service layer instance.

Applications running in the same chip with the service layer can use those interfaces. Same as the APIs, AT commands provide uniform commands about Create, Delete, Open, Close, AddObject, DeleteObject, Notify, UpdateRegister and some URCs. These commands have the same functions as the APIs. Applications that run in the individual MCU and need to communicate with chipset by AT can use these uniform interfaces.

In Device Service layer, it also defines some initial objects, “Security”, “Server”, “Control”, “Device”, “Firmware”, “Location”, “Connectivity Monitoring”, “Connection Statistics”. It provides the definition of the preloaded objects and resources about the device itself.

For the second part, it is the same as LwM2M protocol, it includes four interfaces between LwM2M Server and LwM2M Client.They can be categorized into bootstrap, client registration, device management and service enablement, information reporting, respectively. The Bootstrap Interface is used to provision essential information into the LwM2M Client with which the LwM2M Client can perform the “Register” operation. The Client Registration Interfaces have the function of register, update and de-register. Register is used by the LwM2M Client to register with one or more LwM2M Servers after the bootstrap procedure. Update function is used for each registration to extend the lifetime or add/remove Objects and Object Instances.
De-register is used logout a device when the lifetime of a registration expires. The Device Management and Service Enable Interface is used by the LwM2M Server to access Object Instances and Resources available from a registered LwM2M Client through the use of “Read”,

“Write”, “Execute”, “Create”, “Delete”, “Discover”, or “Write-Attributes” operations. The Information Reporting Interface is used by the LwM2M Server to observe changes in a Resource on a registered LwM2M Client or cancel observation, and used by the LwM2M Client to notify new values when notification conditions configured by “Write-Attributes” operation are met.

The third part is about the Restful API in IoT platform which is the interfaces to AS. It includes four categories: Security Authentication Service, Data Access Service, Data Subscription Service and Data Filtering Service. Security Authentication Service is responsible for authenticating the access requests of AS, obtaining the legality of the token, and refusing illegal requests. Data Access Service deals with legitimate requests for data access from AS, providing interfaces such as “Read Device Resource”, “Write Device Resource”, “Get Resource”, “Create Product” ,“Create device” and “Execute”. Data Subscription Service pushes the data according to the data subscription relationship between clients and AS. “Subscription” and “Data Push Service” interfaces are provided by this service. Data Filtering Service handles the requests for data filtering and cleaning from AS.

6.3. **Resource model**

Both the LwM2M Resource Model and IPSO Smart Object Resource Model [35] evolved from the original concept specified in IETF RFC5988 [36]. This Resource Model has two levels: Object and Resource. The Object IDs and Resource IDs used by LwM2M implementations are defined in the LwM2M Registry [37]. IPSO Smart Objects are registered at that location along with many other Objects defined by other organizations including OMA. When devices use the LwM2M protocol to transfer data, it must conform to those registered Object definitions. China Mobile research institute conforms to the IPSO Smart Objects as registered at the LwM2M Registry for their Communication Suite Implementation for IoT.

IPSO Smart Object Guidelines provide a common design pattern. It is an object model, that can effectively use the IETF CoAP protocol [38] to provide high level interoperability between Smart Object devices and connected software applications on other devices and services.

This object set is intended to be used as a starting place from which to build more as needed. Some of the objects are generic in nature, such as voltage, altitude or percentage, while others are more specialized like the Color Object or the Gyrometer Object. Actuators and Controllers are defined such as timer or buzzer, Joystick and Level. All of these objects were found to be necessary on a variety of use case domains.

The LWM2M defines a simple resource model where each piece of information provided by the LWM2M Client is a Resource. The Resources are further logically organized into Objects. The LWM2M Client can have any number of Resources. Each resources belongs to an Object. In other words, one Object contains a set of Resources. For example, the Firmware Object contains all the Resources used for firmware update purposes. Below Figure illustrates the structure and relationship between Resources, Objects, and the LWM2M Client.

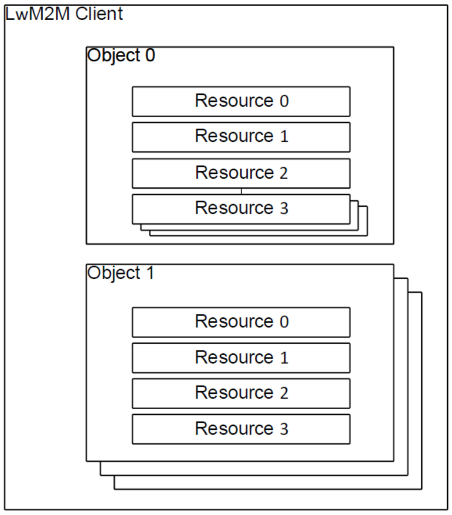


Figure 35 ComSS Resource Model

6.4. LwM2M introduction

LwM2M is a device management protocol that allows the remote manipulation of constrained devices in the Internet of Things. LwM2M uses CoAP, Constrained Application Protocol as a transportation mechanism.

The OMA Lightweight M2M Enabler (LwM2M) is targeted in particular at constrained devices, e.g. devices with low-power microcontrollers and small amounts of Flash and RAM over networks requiring efficient bandwidth usage. At the same time, LwM2M can also be utilized with more powerful embedded devices that benefit from efficient communication. LwM2M provides a light and compact secure communication interface along with an efficient data model, which together enables device management and service enablement for M2M devices.

The LwM2M protocol, to be used for remote management of M2M devices and related service enablement, has at least four outstanding characteristics:

- It features a modern architectural design based on REST appealing to software developers;
- It defines a resource and data model that is extensible;
- It has been designed with performance and the constraints of M2M devices in mind;
- It reuses and builds on an efficient secure data transfer standard called the Constrained Application Protocol (CoAP) that has been standardised by the Internet Engineering Taskforce (IETF) as a variation of the Internet’s HTTP protocol (appropriate for data transfer to and from low-cost connected IoT devices).

LwM2M is an application layer networking protocol for resource constrained devices. It can run on resource constrained Iow devices. On the basis of that, it provides a framework to use the resource models defined by IPSO, reports the scattered data to the IoT platform. The platform can process and analyse data according to the specification of data model defined by IPSO. For operators, it can provide value-added services for the collected data and enhance the value of data.

6.5. LwM2M security

The LwM2M protocol utilizes DTLS with these channel bindings to implement authentication, confidentiality, and data integrity features of the protocol between Client and Server [39]. LwM2M Clients require credentials and configuration information to securely communicate with LwM2M Servers. This configuration information can be provisioned to the LwM2M Client during manufacturing or through the use of the LwM2M Bootstrap-Server. A different set of credentials and configuration information is required in order to secure the communication between the LwM2M Client and the LwM2M Bootstrap-Server.

The security identifiers, endpoint identifiers and keys are used uniformly throughout the LwM2M system to provide a complete security lifecycle solution.

LwM2M supports three different modes of DTLS including Certificates, Raw public keys and Pre-shared secrets.

(1) If a LwM2M server supports X.509 Certificate mode it MUST support the following ciphersuites:

- TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8, as defined in [RFC7251]
- TLS_ ECDHE_ECDSA_WITH_AES_128_CBC_SHA256, as defined in [RFC5289]

If a LwM2M client supports X.509 Certificate mode it MUST support at least one of the cipersuites supported by the LwM2M Server.

Certificate mode of DTLS can be implemented according to [RFC7925], which makes use of the “cached_info” extension. Caching certificate chains allows the client to reduce the communication overhead significantly.

(2) If a LwM2M Server supports the raw public key credentials it MUST support the following ciphersuites:

- TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8, as defined in [RFC6655]
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256, as defined in [RFC5289]

If a LwM2M Client supports the raw public key mode it MUST support at least one of the ciphersuites supported by the LwM2M Server.

(3) A LwM2M Server MUST support the Pre-Shared Key mode of DTLS with the following ciphersuites:

- TLS_PSK_WITH_AES_128_CCM_8, as defined in [RFC6655]
- TLS_PSK_WITH_AES_128_CBC_SHA256, as defined in [RFC5487]

A LwM2M Client MUST support the Pre-Shared Key mode of DTLS with at least one of the ciphersuites specified for the LwM2M Server.

6.6. LwM2M interworking with oneM2M

Interworking between LwM2M and oneM2M is realized through LwM2M IPE, which is composed of LwM2M Server and AE of oneM2M. Through LwM2M-oneM2M interworking, communication between ASN/IN/MN CSEs and LwM2M Endpoints is enabled.

LwM2M IPEs provide the following types of interworking:

(1) Transparent interworking.

The LwM2M IPE encapsulates the LwM2M Objects in Content Sharing Resources and then hosts the Content Sharing Resources in a CSE using the Mca reference points for use by AEs. The AE accesses the Content Sharing Resource from the CSE that hosts the resource using the Mca reference point. Once the AE receives the Content Sharing Resource, the AE extracts the LwM2M Object from the Content Sharing Resource for the AE's purpose.

(2) Translation interworking.

The LwM2M IPE translates the LwM2M Objects into one or more applicable oneM2M Resources and then hosts the resources in a CSE using the Mca reference points for use by AEs. The AE accesses the resources from the CSE that hosts the resource using the Mca reference point.5.3 (LwM2M) IPSO

7. Moving forward with a unified IoT service layer

Projections for massive IoT deployments remain several giant steps away from reality. Acceleration and a unified framework is needed to provide the necessary foundation for all these things and machines to integrate and interact with one another efficiently and effectively.

For a telecommunications operator, deploying a horizontal platform-based framework will address the challenges of fragmentation, integration complexity, information sharing, scalability, operational efficiency and high development cost. An IoT Service Layer based on a horizontal platform brings IoT customers reliable and efficient end-to-end data control/exchange between M2M devices and customer applications by providing functions for remote provision and activation, authentication, data buffering, encryption/decryption, synchronization, aggregation, policy-driven communications and device management. Rapid IoT application development is enabled by the common service functions provided by Service Layer and simple-to use APIs via standardized interfaces.

From a mobile operator’s point of view, the underlying mobile network is an existing large-scale platform for bearing IoT services. Getting the underlying transport network capability to be exposed to IoT applications in a simple way while offering additional and commonly needed function and at the same time guaranteeing a robust protection of the network from inefficient usage will provide differentiated competition for mobile operators' IoT platforms versus other over-the-top offerings.

While, the IoT Service Layer shall be access technology agnostic, it will unlock the huge potential of IoT, bringing the added value by enabling data sharing and efficient network usage between multiple devices, applications, networks and vertical industry segments.

It is a common sense that the cloud side platform of the Internet of Things must allow for vertical sector-specific applications and solutions. We usually call this as “open” platform, i.e. the particular usage of the platform for vertical-specific applications and data models is supported via open and standardized interfaces. In fact, this property of being “open” does not only apply to the cloud side of on IoT Service Layer platform, it is also of high value for implementations of the end-nodes and gateways, which also call for an “open” IoT Service Layer, with openly specified and standardized interfaces to the IoT applications. Enabling collaboration and interworking via standardized interfaces is not only necessary, it is essential.

“Adherence to a single standard will be key in enabling the success of the Internet of Thing on a global basis”[30]. This is also the view of GTI for addressing the challenge of IoT era.

Open Mobile Alliance specifications are used to support management of billions of existing terminals across a variety of wireless network. LwM2M is now attracting attention of mobile operators due to its simple functions to interact between devices and a management server. This concept has been integrated into a bigger context by oneM2M.

oneM2M is a partnership of Standards Defining Organizations (SDOs) developing jointly with more than 200 members a set of specifications that will enable IoT service providers to build a horizontal IoT Service Layer platform, regardless of existing sector or industry solutions. The intention when developing oneM2M specifications is an open and transparent partnership of world-leading SDOs was not to discard or ignore existing industry-specific standards but to work with them to provide added value by extending their reach[12].

oneM2M is scalable, efficient and robust and can be deployed as an overlay to any IP-capable network or with the help of proxies also over non-IP networks, which is going to be good solution for operators building IoT system[31][32]. While essentially being independent of the underlying network technology, it is capable of using MTC and eMTC optimizations of 3GPP-based networks in order to enhance efficiency and offer IoT/M2M related network functions to IoT applications. oneM2M is a well-designed service layer technology, which has the Mcn interface to interwork with one or more underlying network(s) and provides standardized Mca interface to IoT applications. Instantiations of a oneM2M-compliant IoT Service Layer may reside in many of the nodes of an IoT deployment – starting from small sensors or actors up to gateways or infrastructure nodes – in order to act as a “layer” of commonly needed functions for making things and applications interact among each other via an operator’s horizontal platform.

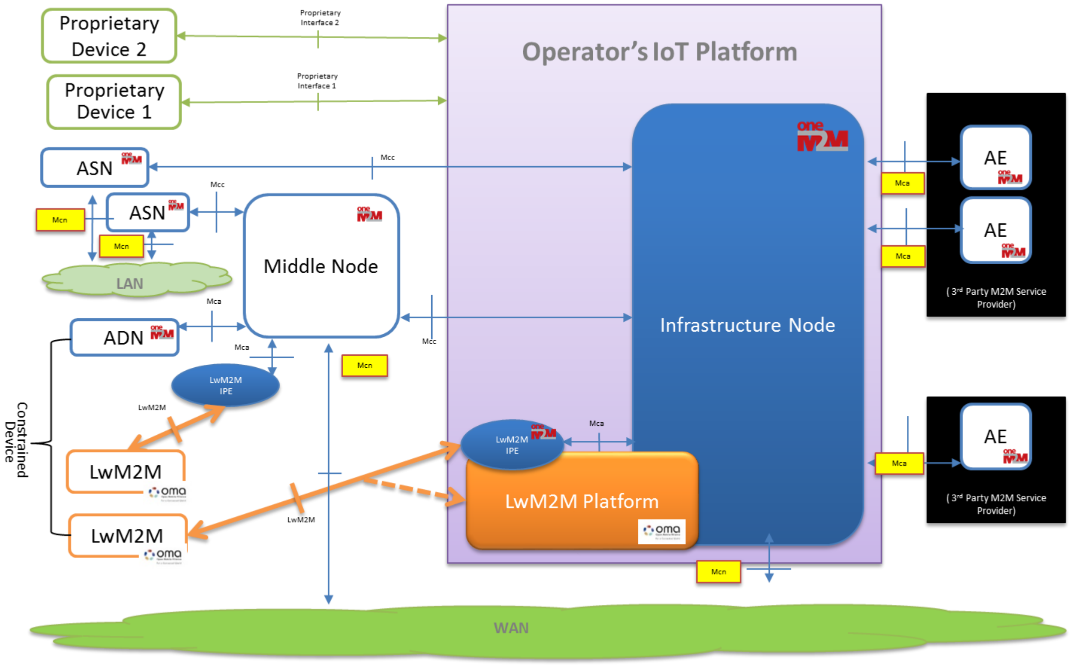


Figure 36 Example oneM2M-compatible IoT service platform with Interworking to LwM2M

Figure 36 depicts an example for building an oneM2M-compatible IoT service platform using the capability of interworking with LwM2M. In the long term, the mobile operators’ IoT service offerings will benefit from oneM2M’s capabilities of interworking with 3GPP network(s), its standardized interface (Mca) to IoT applications for accessing commonly needed functions such as data sharing, as well as oneM2M's standardized interoperability with other technologies, such as LwM2M, OCF, and others.

In summary, deploying a oneM2M-compliant IoT platform that is tightly integrated with a mobile operator's network will allow to address future IoT opportunities in a well-scalable way with a combination of efficient network usage, easy to use IoT APIs, support of secure interactions and data sharing among IoT applications, and management of a large number of devices across a number of different use cases and vertical industry segments.

Reference

[1]. IoT segment: https://iot-analytics.com/top-10-iot-project-application-areas-q3-2016/

[2]. Machina Research report: https://machinaresearch.com/news/press-release-global-internet-of-things-market-to-grow-to-27-billion-devices-generating-usd3-trillion-revenue-in-2025/

[3]. Cisco 2017 Visual Networking Index Report: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.pdf

[4]. oneTransport A transport data marketplace: oneM2M's Role

[5]. Big Data: A Revolution That Will Transform How We Live, Work, and Think; Viktor Mayer

[6]. Machina Research: Research Note Market Shares of LPWA technologies in 2017-2022: trending towards 3GPP, but with considerable fragmentation.

[7]. Analysys Mason, M2M and IoT Opportunities for Telecoms Operators

[8]. Telecoms operators' approaches to M2M and IoT, by Analysys Mason.

[9]. IEC, IoT 2020: Smart and secure IoT platform.

[10].Worldwide IoT Platforms (Software Vendors) 2017 Vendor Assessment – by IDC MarketScape

[11].Thomas R. Eisenmann, Geoffrey Parker, Marshall Van Alstyne, Opening Platforms: How, When and Why?

[12].oneM2M white paper – The Interoperability Enabler for the Entire M2M and IoT ecosystem. http://onem2m.org/images/files/oneM2M-whitepaper-January-2015.pdf

[13].https://www.gsma.com/IoT/wp-content/uploads/2016/11/CLP.26-v1.0.pdf

[14].Market Insight: Designing New IoT Services for IoT Platforms Becomes a Priority – by Gartner, 2017.09

[15].oneM2M TS-0004 Service Layer Core Protocol Specification

[16].oneM2M TS-0001 Functional Architecture

[17].oneM2M TS-0026 3GPP Interworking

[18].oneM2M TS-0003 Security Solution

[19].oneM2M TS-0022 Field Device Configuration

[20].oneM2M TS-0032 MAF and MEF Interface Specifications

[21].oneM2M TR-0008 Security

[22].oneM2M TS0014 LwM2M Interworking

[23].oneM2M TS0021 onem2M and AllJoyn Interworking

[24].oneM2M TS0024 OIC Interworking

[25].oneM2M TS0035 OSGi Interworking

[26].oneM2M TS0023 Home Appliances Information Model and Mapping

[27].oneM2M TS0030 Ontology Based Interworking

[28].oneM2M TS0033 Proximal IoT Interworking

[29].HewlettPackard Enterprise Business White paper: HPE Universal IoT Platform oneM2M and Beyond

[30].UBS investment research report, “Who Are the Enablers of ‘The Internet of Things’?” http://max.book118.com/html/2015/0819/23705417.shtm

[31].SB announced to use onem2M as data collection https://www.softbank.jp/corp/group/sbm/news/press/2017/20170720_03/

[32].M2M Zone, “Korea turns to InterDigital for oneM2M testing”, www.m2mzone.com/ttaint

[33].Francois Ennesser, Wolfgang Granzow: “Security, privacy and device onboarding - The oneM2M approach (based on Release 2A)”, http://member.onem2m.org/Application/documentapp/downloadimmediate/default.aspx?docID=2405

[34].“OMA Lightweight Machine to Machine Protocol v1.0” http://technical.openmobilealliance.org/Technical/release_program/lightweightM2M_v1_0.aspx

[35].IPSO Alliance, http://www.ipso-alliance.org/

[36].RFC 5988 Web Linking https://tools.ietf.org/html/rfc5988

[37].LwM2M Registry, http://www.openmobilealliance.org/wp/OMNA/LwM2M/LwM2MRegistry.html

[38].Shelby, Z., Hartke, K., Bormann, C., and B. Frank,” Constrained Application Protocol (CoAP)”, draft-ietfcore-coap-18 (final), June 2013.

[39].Rescorla, E. and N. Modadugu,” RFC 6347 Datagram Transport Layer Security Version 1.2” January 2012.