ETSI TR 103 843 V1.1.1 (2025-01)

TECHNICAL REPORT

SmartM2M;
oneM2M deployment guidelines and best practices

Reference

DTR/SmartM2M-103843

Keywords

evaluation, KPI, methodology, oneM2M,
performance, scenarios, simulation

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI IPR online database.

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™**, **LTE™** and **5G™** logo are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Technical Report (TR) has been produced by ETSI Technical Committee Smart Machine-to-Machine communications (SmartM2M).

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Introduction

The present document describes the result of TTF 019 (https://portal.etsi.org/xtfs/#/xTF/T019/) whose aims were studying, analysing, evaluating, and simulating IoT application deployments based on some oneM2M open-source implementations. Simulations of the scenarios described in ETSI TR 103 839 [i.1] using the tools developed and described in ETSI TS 103 840 [i.2] and ETSI TR 103 841 [i.3] were conducted and the results described in ETSI TR 103 842 [i.4]. The present document provides qualitative analysis of the results and makes suggestions for additional simulations that can be performed to further characterize a scalable IoT solution using the oneM2M framework.

The redaction of best practices and guidelines for an efficient oneM2M-based IoT solution deployment relies on the analysis of the first results from evaluating open source oneM2M platforms such as OM2M or ACME.

The scope of the present document is to provide oneM2M IoT system designers best practices and lessons learned for the tuning of a oneM2M IoT solution topology, the capacity provisioning, the underlying communication solution and their expected performances.

The work of ETSI TTF 0019 has complemented the ETSI view by liaising with oneM2M technical bodies for consultation during the definition of the deployment scenarios. From the stakeholder point of view, it can help on the adoption of oneM2M through indicators such as performance evaluation to assess the performance of their proposed products. IoT platforms customers can define their specific deployment scenarios and evaluate the performance of a given IoT platform against their scenarios. Finally, the Open-Source Communities around oneM2M may benefit from these results by taking the output of the development of the proof-of-concept simulations into their roadmaps.

1 Scope

1.1 Context for the present document

The oneM2M standard is now mature: multiple deployments exist all over the world at both experimental and operational levels [i.5]. The experimental deployments are conducted for multiple reasons:

- To evaluate the capabilities of the standard in terms of expressiveness, usability on specific equipment, connection with specific existing systems or performance evaluation.
- To provide a methodological study, based on performance evaluation (time, space) on a given set of "paradigmatic use cases".
- To measure KPIs defined in the present document. Different implementations exist that are compliant with the oneM2M standard, available either freely or commercially.

In the present document, use cases are assessed based on selected Key Performance Indicators (KPIs), such as execution time, memory utilization, the quantity of oneM2M entities (e.g. AE, MN-CSE, CSE), data transfer volume, and real-time requirements. Leveraging various available oneM2M Common Services Entity (CSE) implementations, a simulation library or specialized simulator has been developed, enabling performance evaluation and simulation of use cases. This tool provides valuable feedback for system architects, assisting in the selection and fine-tuning of IoT applications within the oneM2M framework. The outcomes of these tool developments and use case evaluations will serve as a foundation for generating additional deliverables. The present document provides the results of an initiative focused on "Performance Evaluation and Analysis for oneM2M Planning and Deployment", addressing five key elements in sequence:

1) A collection of **use cases and derived requirements** were formally identified and defined. This work includes identification of relevant deployment scenarios. The use case style and template from oneM2M was adopted with a minor modification to address some performances issues. This phase of the work resulted in deliverable ETSI TR 103 839 [i.1].

2) The definition of a **performance evaluation model**, with specification of procedures to assess the performance of oneM2M-based IoT platforms. This includes the identification and definition of a set/list of KPIs necessary to assess the deployment. For those KPIs, provision of a formal description of the test campaign and the test results to be obtained. This phase of the work resulted in deliverable ETSI TS 103 840 [i.2].

3) The creation of a **proof of concept** performance evaluation tool. This work also relies on a formal description of the identified deployment scenarios (single vertical domain & multiple vertical domains). This phase of the work resulted in deliverable ETSI TR 103 841 [i.3].

4) A practical **demonstration and analysis** exercise putting the proposed tool to use, with a specific oneM2M implementation but aimed at being a blueprint for the adoption and re-use of the results of ETSI TR 103 839 [i.1], ETSI TS 103 840 [i.2], and ETSI TR 103 841 [i.3] with other oneM2M implementations and deployment scenarios. This phase of the work was used in deliverables ETSI TR 103 842 [i.4] and ETSI TR 103 843 (the present document).

5) The development of a set of **guidelines and best practices** documenting best practices and lessons learnt as well as providing instructions for IoT solution topology, capacity provisioning, and expected performances that will gives crucial directives and information to designer and implementors. This phase of the work resulted in ETSI TR 103 843 (the present document).

The present document covers the last of the five items listed above and relies on the content of the related ETSI publications listed below:

- ETSI TR 103 839 [i.1].
- ETSI TS 103 840 [i.2].
- ETSI TR 103 841 [i.3].
- ETSI TR 103 842 [i.4].

1.2 Scope of the present document.

The present document is structured as follows:

- Clauses 1 to 3 set the scene and provide references as well as definition of terms, symbols and abbreviations, which are used in the present document.
- Clause 4 describes observations regarding deployment options offered by oneM2M and recommendations related to selection criteria for which oneM2M features to use.
- Clause 5 describes the benefits of simulation in the deployment of large scale IoT deployments.
- Clause 6 provides the conclusions of this work.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI TR 103 839: "SmartM2M; Scenarios for evaluation of oneM2M deployments".

[i.2] ETSI TS 103 840: "SmartM2M; Model for oneM2M Performance Evaluation".

[i.3] ETSI TR 103 841: "SmartM2M; oneM2M Performances Evaluation Tool (Proof of Concept)".

[i.4] ETSI TR 103 842: "SmartM2M; Demonstration of Performance Evaluation and Analysis for oneM2M Planning and Deployment".

[i.5] oneM2M: "List of oneM2M deployments".

[i.6] Tools Deliverable ETSI repository linked with ETSI TR 103 841.

[i.7] ETSI TS 118 123: "oneM2M; SDT based Information Model and Mapping for Vertical Industries".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

guidelines and good practices: methodological document that gives hints to deploy a oneM2M infrastructure

Key Performance Index (KPI): list of criteria to be measured on a system

oneM2M: ETSI standards for M2M and the Internet of Things

NOTE: www.oneM2M.org is an ETSI Partnership Project (EPP) on M2M launched by a number of SSOs including ETSI.

oneM2M deployment: mapping of IoT applications on a oneM2M infrastructure

oneM2M standard implementation: implementation of the oneM2M standard

performance evaluation: evaluation of temporal, data transfer volumetry and scalability aspects of a system

platform evaluation tool: simulation environment that is used to calculate/demonstrate the performance of a system

profiler: monitoring tool measuring KPIs

real time constraints: dynamic constraints to be fulfilled related to time

single/multiple horizontal/vertical domains: interaction capability of many oneM2M infrastructures from different domains

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ACME An open source CSE Middleware for Education

NOTE: It is available at https://acmecse.net/.

AE Application Entity
API Application Program Interface
ASN-CSE Application Service Node Common Service Entity
BACnet Building Automation and Control networks
BLE Bluetooth Low Energy
CBOR Concise Binary Object Representation
CoAP Constrained Application Protocol
CPU Central Processing Unit
CSE Common Service Entity
DECT NR+ Digital Enhanced Cordless Telecommunications New Radio plus
eMTC enhanced Machine Type Communication
EPP ETSI Partnership Project
ETSI European Telecommunications Standards Institute
HTTP Hyper Text Transfer Protocol
IN-CSE Infrastructure Node - Common Services Entity
IoT Internet of Things
JSON JavaScript Object Notation
KPI Key Performance Index (term)
KPI Key Performance Indicator (abbreviation)
LoRa Long Range (wireless technology)
LTE-M1 Long-Term Evolution category M1 (LTE Cat-M1)

NOTE: "enhanced Machine Type Communication" (eMTC) is known as LTE Cat-M1.

M2M Machine-to-Machine
Mca Reference Point for M2M Communication with AE
Mcc Reference Point for M2M Communication with CSE
MN-CSE Middle Node - Common Services Entity
MQTT Message Queuing Telemetry Transport
NB-IoT NarrowBand Internet of Things
OM2M Eclipse OM2M - Open-Source platform for M2M communication

RTT	Round Trip Time
SSO	Standard Setting Organization
TR	Technical Report
TS	Technical Specification
Wi-Fi®	Wireless Fidelity

NOTE: IEEE 802.11™ family of standards.

XML	eXtensible Markup Language

4 Lessons learned from use cases simulations

4.0 Foreword

The present document aims to examine the key features and capabilities of oneM2M that must be addressed when scaling an IoT deployment. The deployment process begins by establishing the foundational features and functionalities of the intended application or service. Three example use cases were analysed, providing a framework that can be adapted for other types of deployments. In any successful IoT rollout, the critical next step following initial deployment and proof of concept is expanding the system to accommodate larger-scale requirements.

The oneM2M standard offers many features and capabilities that facilitate and reduce the engineering and architectural effort needed to scale an IoT deployment. The oneM2M CSE makes application scaling easier in terms of support for more devices, integration of a heterogenous devices, discoverability of the devices and data generated by the devices, and federated deployment and multitenant access control and sharing of data.

The tool developed enables the simulation of several core oneM2M features, facilitating analysis to support implementation decisions, capacity planning, and the exploration of alternative oneM2M features for potential use.

Some critical considerations include:

1) Protocol selection. The oneM2M standard supports communication with the CSE using HTTP(s), CoAP(s), MQTT(s), and websockets. The simulation model can define the protocol being used at each connection to the CSE. For example, the device communicating to the CSE may use CoAP(s) while the application may communicate using websocket protocol. Using the simulation the architect can explore using MQTTs instead of CoAP(s).

2) CSE to device communications. IoT deployments may primarily entail the device sensing data and sending that data to the CSE for processing and application use. Even in these single direction communication structures, as the IoT system moves beyond initial proof of concept and scales to larger deployments, the need to send data to the device becomes apparent. This communication may include device configuration parameters, security credentials, and remote firmware updates. A oneM2M CSE offers multiple approaches to support this bidirectional communication. These approaches can be modelled in the simulation and analysed in the context of future or current implementation choices.

3) Communication channel. IoT systems are effectively deployed in many types of environments. Some examples are in cities, farms, industrial factories, highways, office buildings, ocean liners or ships, etc. In many of these environments the selection of the communication channel can directly impact on the cost of deployment. Some examples are the cost of the communication module and the engineering effort to integrate the communication module. Example communication module choices are cellular such as NB-IoT or LTE-M1, Wi-Fi®, LoRa™, BLE and DECT NR+, just to mention a few.

4) Power source. IoT deployment decisions are heavily impacted by the source of power that will be available to the device. The design choices available for a battery powered device located at a fixed position in a remote area such as a farm are much different than the options available to devices mounted on a traffic light in a city. While the simulation does not directly simulate power consumption, it models CPU utilization which can be correlated with power consumption. For example, an IoT device that is required to communicate multiple times per second to the oneM2M CSE will be much different than a device that only needs to communicate one time per day.

5) Multi-Tenancy. Successful IoT deployments will eventually need to consider multi-tenancy solutions. One of the key features of the oneM2M standard is the ability to communicate across silos or verticals, which can also be viewed similarly to different customers. There are options for how to implement the ability to communicate across these logic data boundaries in oneM2M. One option is to have separate oneM2M CSEs for each vertical or customer. Another option is to use a single oneM2M with appropriate access control policies (<accessControlPolicy>). Data can be shared using oneM2M procedures for retargeting requests or announcing resources to remote CSE. Using the simulator described in ETSI TR 103 841 [i.3] these options can be evaluated.

4.1 End devices: number and associated velocity

4.1.1 Description

Simulation of an IoT deployment follows the same development process as the actual deployment. First start with a simple characterization of the deployment, as shown in Figure 1.

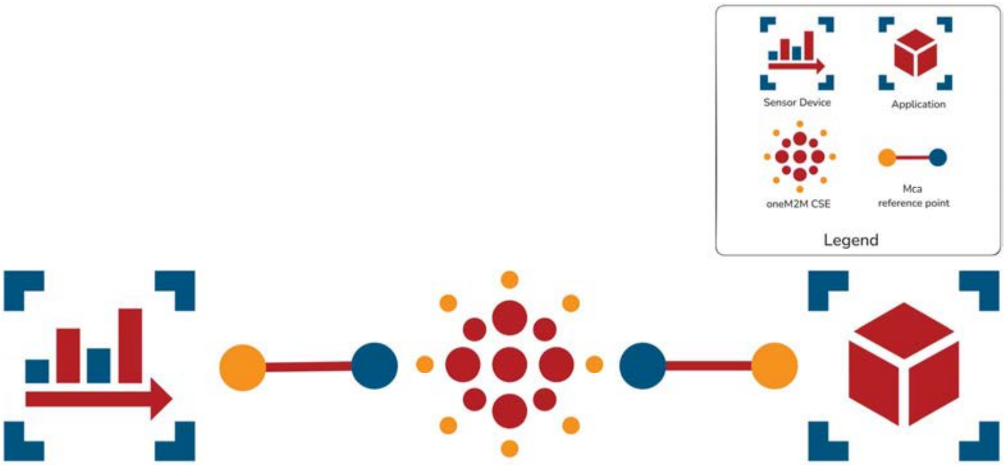

Figure 1: Simple oneM2M deployment

At this stage the goal is to capture the basic characteristics of the communication of the device, including the size of the payload, the frequency of the communication to the oneM2M CSE and the underlying protocol used. Similar information is needed for the application communication to the CSE. In the case of the application, it is already possible to start considering which oneM2M features to use. For example, the application can periodically retrieve the data from the device's data model or alternatively a subscription can be modelled so that whenever the device sends new data the oneM2M CSE sends the data to the application. An example of this comparison is shown in clause 7 of ETSI TR 103 842 [i.4].

4.1.2 Impacted KPIs and Potential bottlenecks

From a developmental perspective, it is quite easy to implement periodic polling of the oneM2M CSE to retrieve the current state of the desired data resource. Then, in the simplest implementation the view of the new data is updated. In a live / dynamic view of moderate to large amounts of data that are changing frequently this may be the best approach in terms of complexity for development, amount of payload transmitted and latency of the system. Alternatively, if the data is slowly changing or the data requires multiple requests to get all the desired data for a view the use of the oneM2M subscription services may be the best approach. For example, if the data model uses separate containers for each type of data from the sensor device, or separate devices provide the data that is used in a single view on the application then polling for multiple separate data resources presents a system that is not scalable and will quickly experience bottlenecks from the many application requests. The oneM2M subscription process will send a resource that has changed to all subscribers when the change occurs. This leads to a significant reduction in requests that the oneM2M CSE has to handle that in some cases may include responses that were the same as a previous response.

An example of this is shown in clause 7 of ETSI TR 103 842 [i.4] where periodic polling with different rates of polling frequency is simulated. The results, replicated here, show that the more frequent the polling frequency (better latency with respect to showing the latest data) the more resources consumed by the CSE.

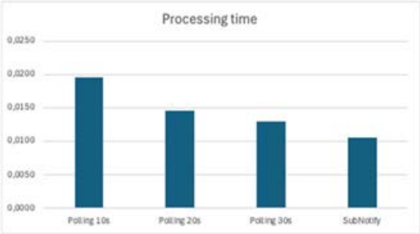

Figure 2: Processing time for traffic light Use Case

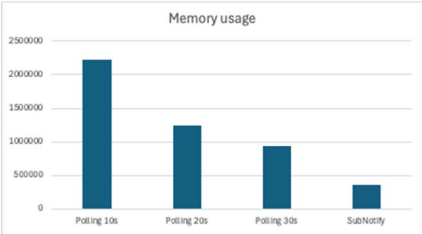

Figure 3: Memory usage for traffic light Use Case

4.1.3 General Recommendations

Recommendation #1: Prefer the oneM2M subscription and notification process over periodic polling.

Exception: Battery powered Cellular IoT devices are best operated with periodic polling rather than having to maintain an active connection since operation of the cellular modem in this manner can represent a very large portion of the power budget and capacity of the battery.

Recommendation #2: Prefer a data model that places all the data used by an application into a single resource.

Exception: When the data is slowly changing, the benefit of grouping all the data together is reduced compared the impact of having a data model that is not logical or too large.

4.2 Underlying CSE architecture support

4.2.1 Scale the number of CSE according to device environment

For small deployments a single CSE that meets the desired throughput and latency requirements is sufficient. However, as the system grows to meet additional requirements, the capabilities of the CSE to operate on different platforms and under different functional or organizational controls becomes a bigger part of the deployment considerations. Just as the deployments need to scale, the simulation can be scaled to model the aggregated throughput, latency and processor utilization of each node.

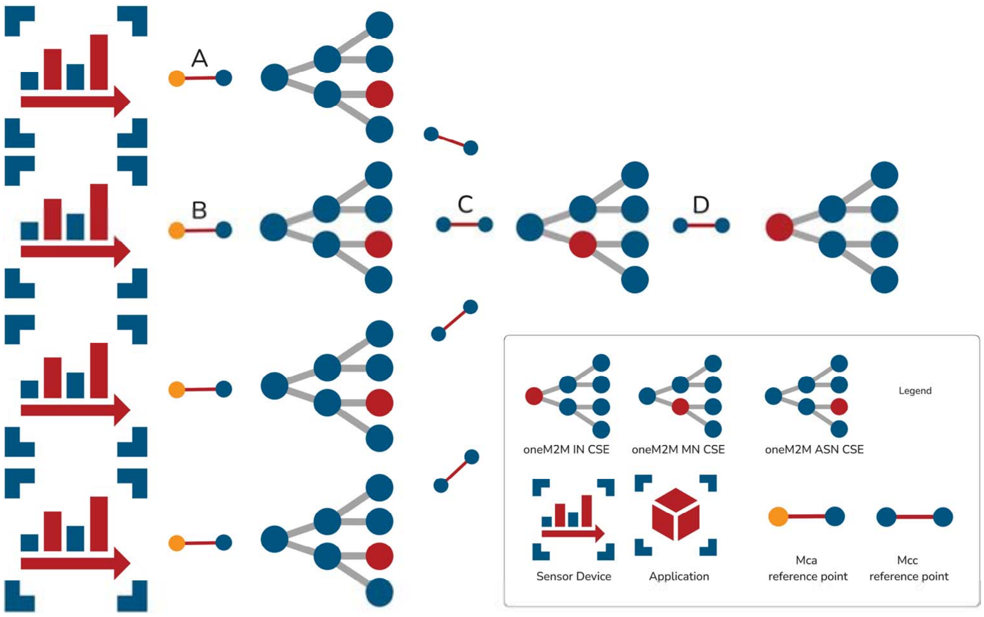

Figure 4: Scaling of oneM2M CSEs

An example of this is shown in clause 6 of ETSI TR 103 842 [i.4]. In Figure 4 the simulation capabilities of the oneM2M deployment are depicted. The model allows for specifying different metrics for each processor hosting a CSE or the same metrics for all of them. The model also allows for modelling different communication channels between each node. For example, a first device may communicate with the ASN-CSE using Wi-Fi® while another device may communicate using BLE. The ASN-CSEs may communicate to the MN-CSE using a cellular channel and the MN-CSE to IN-CSE may communicate using ethernet. Similarly, each device may have different payload types, such as XML, JSON or CBOR. While the simulation does not directly model these types, they can be indirectly modelled using the payload size.

4.2.2 Impacted KPIs and Potential bottlenecks

Scaling the IoT deployment has multiple considerations. A first option to scale the system is simply a higher capacity host processor. This is by far the easiest way to scale a system with well-known capacity needs, and it does not require any additional design or development work. A second option to scaling a oneM2M system is horizontal scaling of the components of the system. This requires a thorough understanding of the sub-component performance measures. For example, if the database size is a limit, select a database that supports sharding to increase the ability to handle more data. Alternatively, if the CSE implementation can support scaling such that operations that take a long time can be parallelized, multiple instantiations can achieve more capacity. A third option for scaling a oneM2M system is federation of the oneM2M CSEs, which is a feature set that is integral to the oneM2M architectural definition. This takes advantage of the ability to register CSEs in a hierarchy and provisioning the devices to communicate with the appropriate CSE. The provisioning can be based on vertical basis, such as traffic lights communicating with one CSE and emergency vehicles communicating with another CSE. When scaling in this manner, the data on one CSE can be made accessible to other CSEs by retargeting requests or using the "announceTo" feature of oneM2M to expose the data and keep it synchronized with the original source of data.

All these options for scaling can be simulated using the tools and methods described in ETSI TR 103 841 [i.3]. These options can also be combined. In clause 6 of ETSI TR 103 842 [i.4] a simulation that models the impact of using different capacity host processors is explored. The results, replicated here, show that the round-trip time of primitives improves as the host processor is upgraded.

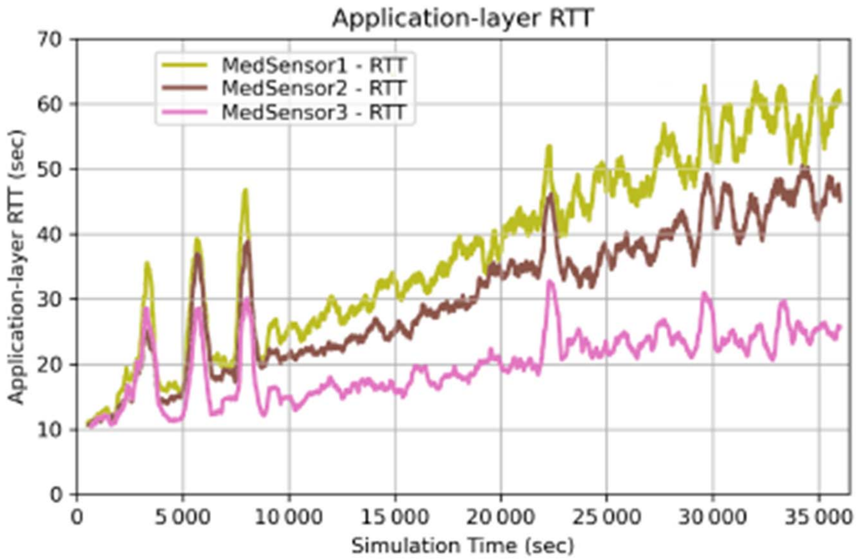

Figure 5: Round Trip Time (RTT) of primitives as Host Processor is scaled

4.2.3 General Recommendations

Recommendation #3: Prefer to scale the host processor first. This is the fastest and easiest approach to achieving greater performance of a specific node in the deployment.

Exception: None.

Recommendation #4: For large scale IoT deployments consider deploying the CSEs in a hierarchy that models the business use cases. For example, if the same system is deployed to two factories, devices from factory 1 should communicate with a CSE dedicated to factory 1 and devices from factory 2 should communicate with a second CSE dedicated for factory 2.

Exception: This separation should not be applied at too fine level of granularity. If the two CSEs have frequent communications between each other in order to function properly, the gains from separating the functionality can be lost in the extra communication between the two CSEs. This was not one of the examples modelled but the simulator does support this type of analysis.

Recommendation #5: Data that must be shared to another CSE should use the "announceTO" functionality of oneM2M to replicate the data on the remote CSE. This reduces the latency of the request for the data by the data consumer.

Exception: If the data to be accessed is rarely used, it may be better to use retargeting to request the remote data. It reduces the network traffic and processing utilization required to keep the replicated data synchronized.

4.3 Number and Diversity of network technologies

4.3.1 Impact of technologies on end-to-end delay

The choice of technology for communication is important for any IoT deployment. However, in a oneM2M deployment, communication technology is not a significant consideration in terms of how to deploy oneM2M CSEs or how to create the data model that will be used. The selected technology does require consideration of which oneM2M features to use. In clause 5 of ETSI TR 103 842 [i.4] the smart campus scenario used cellular network, BACnet, Wi-Fi®, ethernet and fibre channel. The simulation tool does not select any features or services based on communication technology, but it does allow different metric definitions depending on the technology. For example, referring to Figure 6, communications to the MN-CSE using BACnet will experience different latencies than the ethernet connection to the IN-CSE. Also, it should be highlighted that the application using data from all the various devices using a heterogenous selection of communication technologies is completely agnostic to those communication technologies.

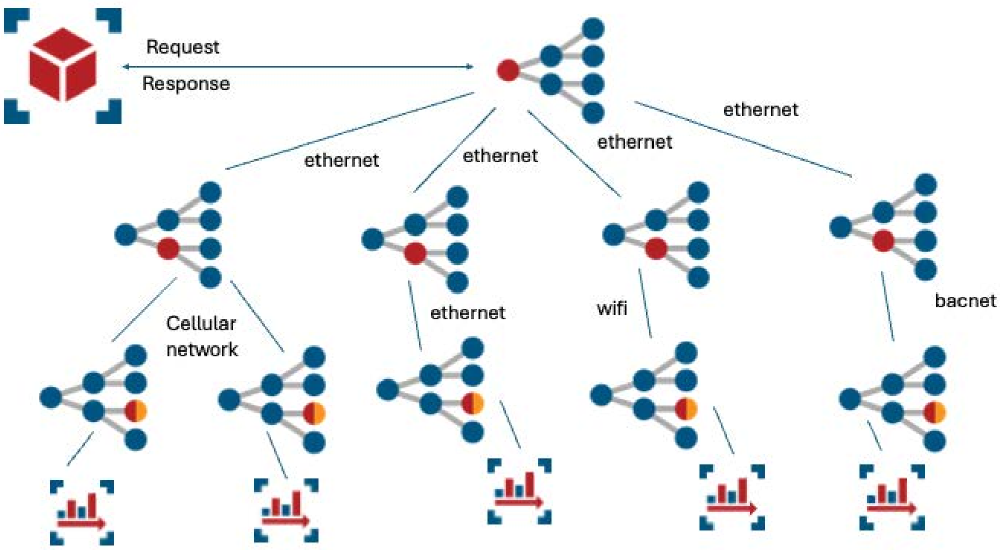

Figure 6: Comparison of different localization of AE

4.3.2 Impacted KPIs and Potential bottlenecks

Aspects related to the communication technology should be considered separately from the data that the device provides. For example, a sensor may have two communication models on its host processor such as BLE and LTE-M1. In the oneM2M resource tree, connection details for both channels should be kept separate from the data provided by the devices. This will allow the data to be shared without consideration of the communication methods used. In Figure 6, if all the sensors provide temperature measurements, the applications that use the temperature values do not have insight into methods of transporting the data.

The performance of the overall system can be impacted by the communication methods deployed. The differences in latency and round-trip time can affect the responsiveness of the user application and in some cases cause the basic system requirements to be missed. For example, if a presence detector has a requirement to report detected movement within 2 seconds, the tools developed here can show the system latency based on the selected communication protocols, selection of polling rates or subscribe/notify features. It can also show how the selection of cellular communications versus BLE impact the latencies of the messages.

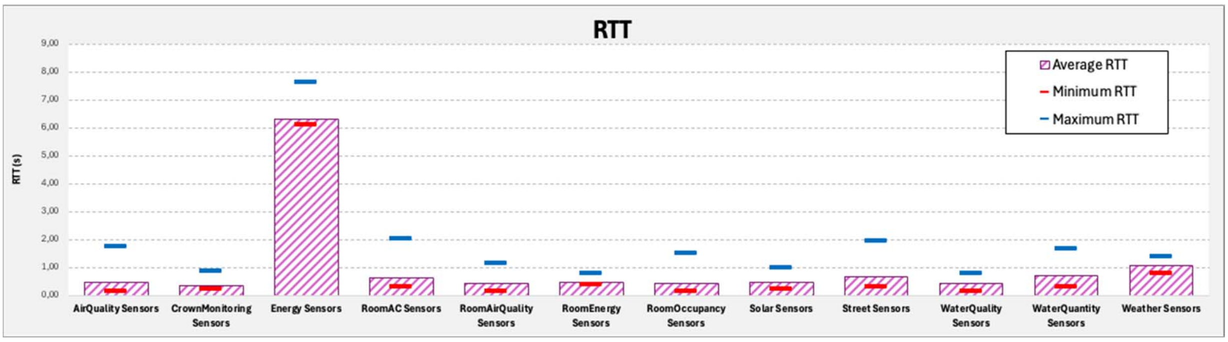

Figure 7: Simulated response time for the different families of sensors with centralization on IN-CSE

The simulator can be used to show the expected performance of the aggregated system. Figure 7, extracted from simulation results described in clause 5 of ETSI TR 103 842 [i.4], shows the impact of the communication methods for several sensors that use different communication technologies.

4.3.3 General Recommendations

Recommendation #6: Measuring performance at large scale should be done with simulation. For any IoT system, simulation can be a more cost and time effective solution for estimating the impact of different technology solutions. Simulations should include both the numbers of devices in the system as well as the number of applications in the system. A system with a single application that makes 3 API calls per second might be acceptable, but when there are many instances of that application the capabilities of the network can be challenged.

Exception: If the actual systems being considered do not have accurate performance metrics and measurements then the accuracy of the simulation will be reduced, possibly leading to significant underestimation of the infrastructure needed to support the system.

Recommendation #7: Capture actual process metrics, using the tools described in ETSI TR 103 841 [i.3] regularly. Additionally, an IoT deployment should have a formal metrics collection plan that runs all the time. This allows for accurate analysis of current operations, and historical data that can be used for fault analysis and future growth needs.

Exception: None.

Recommendation #8: Prefer IoT system designs using event driven architectures to allow for easier modification of communications channels used. During the evaluation of different design choices that can be evaluated using the present simulation tools, it becomes apparent that one solution or design choice is not necessarily good for the next design choice. This can be true of multiple components in the same deployment. For example, a security system that has motion sensors may use cellular communications at the far edges of the property monitored while the motion sensors located on the front door may use Wi-Fi®. The underlying sensor logic is the same, only the communication module changes, so it makes sense to design this in a modular manner that is easy to modify. A similar example can be described for the selection of the data model used and the method implemented for bi-directional communications. In all cases if the design is modular and event driven then application logic is simplified and made more robust.

Exception: This recommendation has the most value for greenfield or when significant architecture modifications are occurring. It does not have as much validity when the system is mostly complete using a different architecture as the effort may exceed the benefit. It should be noted that this recommendation is subjective based on the simulation choices that can be made, but there are no metrics available from the present tool to support this recommendation.

4.4 oneM2M data model and communication methods

4.4.1 Description

The oneM2M service layer offers different communication possibilities in terms of the communication channels described above, but also in the method of representing the data in the resource tree. For example, the Smart Campus scenario may represent data from sensors using any of several different data models available in oneM2M.

Approach #1 - Custom Data model: Represent all data from a sensor in a single <contentInstance> resource where all the data from the sensor is available in a proprietary manner in that resource.

Approach #2 - Custom Data model: Represent each data type in a separate <contentInstance> resource.

Approach #3 - Custom Data model: Represent each data type in a custom <flexContainer> resource.

Approach #4 - Standard Data model: The Smart Device Template specifications defined in ETSI TS 118 123 [i.7] describe data models for many devices across several domains.

4.4.2 Impacted KPIs and Potential bottlenecks

The selection of the data model can have significant impact on the complexity of implementation, power budget of battery powered systems, bandwidth of densely populated systems and computational utilization of the host processor of the oneM2M CSE. Also, the selection of opaque data sharing resources such as the <contentInstance> and <timeSeriesInstance> can limit the available oneM2M features that an application can use.

Each approach has implications for the system architecture in terms of complexity to implement at the device as well as complexity at the application. For example, some data models can be populated with a single oneM2M primitive while others may require more primitives depending on the number of sensors or characteristics of the device. This impacts the number of primitives required as well as the size of the payload when transmitted. These two factors have implications for the chosen communication method. There is a similar consideration for the application that uses this data. In some options the application needs to be aware of the proprietary data structure and the ability to parse the payload while in other options the payload is not opaque and can easily be understood semantically. This impacts the amount of data transmitted and the complexity of the application using the data.

Another related consideration is how the application is made aware of new data. In approach #1 above if the data model is represented as a collection of multiple values in a <contentInstance> the application must analyse each new <contentInstance> to determine what has changed (if this is important to the application). In the other approaches the change can be determined by the oneM2M CSE on behalf of the application, reducing the frequency of primitives and the payload of the messages.

4.4.3 General recommendations

Recommendation #9: Perform system simulations using as much detail as possible. Use the simulation process as an aid to understand both low level and high-level aspects of the systems performance capabilities and limits. Iterate the simulations by adding details as they are considered. For example, a first simulation iteration could model the device communications without deciding on the data model and simply specifying a payload and frequency of measurements. A second iteration can add more details such as the specifics of using a <contentInstance> or a <flexContainer>. A third iteration can add additional details that only transmit the <flexContainer> attributes that have changed. With each of these iterations, many more questions about the system can be answered as well as new questions being raised. Addressing questions like this is much more efficient in simulation than in actual prototype development.

Exception: None. If development time is constrained, simulation is likely to be eliminated however this in most cases introduces technical debt.

Recommendation #10: Use oneM2M service layer features or plan to use them in the future. The first implementation is likely to have a limited scope and not require many of the features that oneM2M exposes for application development. However, as the system grows to accommodate more applications and more devices the architecture must reduce application complexity. This includes addressing the number/frequency of primitives and reducing unwanted overhead from the content of primitives. An example for IoT devices that is a key enabler for this type of growth is proper device management with emphasis on firmware update capabilities. There are many other features that go beyond the scope of the present document. Refer to www.oneM2M.org for more information.

Recommendation #11: Use oneM2M <subscription> resource to receive notifications when resources change to reduce primitive frequency. The capabilities exposed by the oneM2M <subscription> resource are quite extensive. The base capability alone can significantly reduce the messaging in the system. This is critical for a scalable application deployment. Beyond that the <subscription> resource can be configured to apply thresholds to the resource attributes to further limit when notifications are generated. There are many more features of the <subscription> resource that are powerful enablers for simplifying an application and reducing the network utilization.

Exception: None.

Recommendation #12: Use data models consisting of <contentInstance> or <timeSeriesInstance> as a last option rather than a first option. Many of the oneM2M features depend on being able to process the data in a resource. The <contentInstance> resource and the <timeSeriesInstance> resource are considered opaque and do not support many of these features.

Exception: Migration from a legacy IoT deployment to a oneM2M deployment is a good use for these opaque data models. Using these two resources can be the quickest way to migrate an IoT device to a native oneM2M device.

5 Experimenting lessons learned in the developed tools

5.1 Benefits of Simulation

Simulations offer a powerful way to optimize IoT system design, deployment, and scaling by providing insights into system behaviour under various configurations and workloads. Whether matching planned deployments, analysing current deployments, or exploring potential deployment options, simulations serve as a critical tool for enhancing system performance and efficiency. Several key benefits of IoT simulations are:

- **Resource Optimization.** Developing and deploying an IoT system involves many interconnected components that work together to deliver value to customers and end users. Simulations allow system designers to test configurations and identify potential bottlenecks or inefficiencies before committing significant resources to physical deployments. This proactive approach saves time, reduces costs, and ensures the effective allocation of hardware, software, and network resources.
- **Risk Reduction.** By modelling various scenarios, simulations can identify vulnerabilities, scalability issues, and performance constraints. This mitigates the risks associated with deploying systems that may underperform or fail under real-world conditions. Simulations also provide a safe environment to test new configurations or strategies without disrupting active deployments.
- **Iterative Refinement.** Building reliable simulations, like software development, is an iterative process. Starting with smaller, foundational models, system designers can refine their simulations based on results and feedback, gradually adding more precision and detail. This iterative approach ensures continuous improvement and adaptation as system requirements evolve.
- **Scalability Testing.** Simulations enable designers to test the scalability of IoT systems by emulating large-scale deployments with varying numbers of devices, nodes, and workloads. This helps validate the system's ability to handle growth while maintaining performance, reliability, and security.
- **Fidelity and real-world validation.** Once decisions are made based on simulation results and the modelled system is deployed, it is essential to repeat profiling measurements using the deployed system. This feedback loop improves the simulation's fidelity and accuracy, providing deeper insights into how the system will respond to future changes or growth.
- **Flexibility and future planning.** Simulations provide a platform to experiment with new deployment options, configurations, and technologies. This flexibility helps organizations adapt to changing market demands, integrate emerging technologies, and prepare for future expansions.

5.2 Best Practices for IoT Simulations

Recommendation #13: Leverage Advanced Tools such as those described in ETSI TR 103 841 [i.3]. Simulation tools are designed to reduce the initial investment required to create system simulations. By leveraging these tools, organizations can accelerate the simulation process, focus on refining models, and explore more scenarios without significant overhead.

Recommendation #14: Start small and scale gradually. Like deploying any large software project, simulations should begin with small, foundational systems. Based on insights from initial tests, designers can incrementally add complexity and detail, ensuring manageable progress and continuous learning.

Recommendation #15: Close the loop between simulation and deployment. Treat the simulation and deployment process as a continuous cycle. Simulations inform deployment decisions, and real-world deployments provide data to refine simulations. This synergy improves both the accuracy of future simulations and the reliability of deployed systems.

Recommendation #16: Focus on measurable KPIs. Clearly define Key Performance Indicators (KPIs) for the simulation, such as latency, throughput, memory usage, and scalability limits. Use these metrics to evaluate system performance and guide optimization efforts.

6 Conclusions and follow-up

The present document offers a comprehensive guide on leveraging oneM2M for IoT deployments, focusing on the importance of simulations for performance evaluation, capacity planning, and scalability analysis. It emphasizes that the oneM2M standard has matured, with real-world and experimental deployments validating its capabilities. The simulation tools described in ETSI TR 103 841 [i.3] are based on Key Performance Indicators (KPIs) such as latency, memory usage, and data throughput, which are critical for analysing deployment scenarios.

Key takeaways include:

- Simulation Benefits: Simulations help identify system bottlenecks, optimize resource allocation, and evaluate scalability, ensuring efficient IoT system design and deployment.
- Use Case Insights: Several scenarios were explored (e.g. smart campus, eHealth, traffic lights) to illustrate how simulations support architectural decisions and system performance enhancements.
- Critical oneM2M Features: Subscription mechanisms, hierarchical CSE deployment, and modular communication design are highlighted as enablers of scalability and efficiency.
- Simulation Tools: The tools enable testing of diverse configurations and deployment options, such as protocol selection, communication technologies, and data modelling, reducing risk and improving system design accuracy.

Simulating large IoT deployments is more than a preparatory exercise; it is an ongoing process that enhances decision-making, reduces risks, and ensures system resilience in a rapidly evolving technological landscape. By embracing simulations as an integral part of IoT system development, organizations can deliver robust, scalable, and customer-centric solutions.

History

Document history		
V1.1.1	January 2025	Publication