ETSI TR 103 839 V1.1.1 (2023-09)

TECHNICAL REPORT

SmartM2M;
Scenarios for evaluation of oneM2M deployments

Reference

DTR/SmartM2M-103839

Keywords

IoT, 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 information pertaining to these essential IPRs, if any, is 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 Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, 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.

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.

Executive summary

The present document proposes three use cases scenario of IoT systems specially selected and defined for representing situations where a modelling and an analysis of the application performances is mandatory. These uses cases are compliant with the oneM2M standard. They will serve to define a data and a behavioural model for an evaluation of the application but also the deployment on a oneM2M implementation. The first use case is currently deployed in a smart campus environment, the second is a generic one that highlights IoT application with event-triggered and time-triggered characteristics. The last one is a traffic light control system with synchronization features. For all of the scenarios specific temporal and behavioural constraints to be verified and Key Performance Indexes (KPIs) that have to be verified are identified.

Introduction

The objective of the present document in conjunction with three other ones [i.2], [i.3] and [i.4] is to provide three use cases that articulate across different verticals and identify a list of common requirements across these verticals and to reason on it with a perspective of simulation and performance evaluation.

1 Scope

1.1 Context for the present document

The oneM2M standard is now mature and multiple deployments exist all over the world at both experimental and operational levels. 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". The present document will evaluate use cases in terms of running time, memory space, numerosity of oneM2M entities (like e.g. AE, MN-CSE, CSE, etc.), data transfer volume and real-time needs. The present document will use a selection of available oneM2M CSE implementations. The present document will provide a tool to evaluate and simulate the performance of the use cases. The results of this tool development and evaluations of the use cases will be the basis to generate three Technical Reports [i.2], [i.3] and [i.4] and one Technical Specification [i.1]. The present document was developed in the context of ETSI Testing Task Force (TTF) T019, set up to perform work on "Performance Evaluation and Analysis for oneM2M Planning and Deployment". Five elements were addressed sequentially:

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

2) The definition of **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.1].

3) The creation of a **proof of concept** of a 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 ETSI TR 103 841 [i.2].

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 (the present document), ETSI TS 103 840 [i.1] and ETSI TR 103 841 [i.2] with other oneM2M implementations and deployment scenarios. This phase of the work resulted in ETSI TR 103 842 [i.3].

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, expected performances. This phase of the work resulted in ETSI TR 103 843 [i.4].

The present document covers the first of the five items listed above and provides the basis for the related ETSI publications listed below:

- ETSI TR 103 839: Scenarios for evaluation of oneM2M deployments (the present document);
- ETSI TS 103 840: Model for oneM2M Performance Evaluation [i.1];
- ETSI TR 103 841: oneM2M Performances Evaluation Tool (Proof of Concept) [i.2];
- ETSI TR 103 842: Demonstration of Performance Evaluation and Analysis for oneM2M Planning and Deployment [i.3];
- ETSI TR 103 843: oneM2M deployment guidelines and best practices [i.4].

1.2 Scope of the present document

The present document identifies additional requirements to be potentially submitted to oneM2M in the areas of performance evaluation by means of relevant use cases. The oneM2M architecture ([i.5] and [i.6]) is targeted as the standard to be evaluated regarding these use cases and requirements. 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 the method used for collecting Use Cases providing a proposal of a tiny extension of the use case "template" as provided in oneM2M-Template-Use-Case [i.7] and extensively used in oneM2M TR-0001 [i.8].
- Clause 5 presents Use Case, called "Smart Campus".
- Clause 6 presents a Use Case, called "Generalization of event trigger/periodic event IoT system" that focuses on classical event triggered and or time triggered behavioural aspects to be captured in IoT systems. The objective is to obtain, based on such an example, some generic constraints to be expressed and to determine the associated constraints and KPI to be evaluated.
- Clause 7 presents a Use Case, called "Traffic lights Control and monitoring", reflecting a deployment by C-DAC, India and several generic implementation variations.
- Clause 8 presents oneM2M feature coverage analysis of the use cases collected.

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 TS 103 840: "SmartM2M; Model for oneM2M Performance Evaluation".

[i.2] ETSI TR 103 841: "SmartM2M; oneM2M Performances Evaluation Tool (Proof of Concept)".

[i.3] ETSI TR 103 842: "SmartM2M; Demonstration of Performance Evaluation and Analysis for oneM2M Planning and Deployment".

[i.4] ETSI TR 103 843: "SmartM2M; oneM2M deployment guidelines and best practices".

[i.5] oneM2M TS-0001 (V4.19.0): "Functional Architecture".

[i.6] ETSI TS 118 111 (V2.4.1): "oneM2M; Common Terminology (oneM2M TS-0011 version 2.4.1 Release 2)".

[i.7] oneM2M-Template-Use-Case.

[i.8] oneM2M TR-0001 (V4.3.0): "Use Cases Collection".

[i.9] Shubham Mante, SVSLN Surya Suhas Vaddhiparthy, Muppala Ruthwik, Deepak Gangadharan, Aftab M. Hussain, Anuradha Vattem: "A Multi-Layer Data Platform Architecture for Smart Cities using oneM2M and IUDX", 8th IEEE World Forum on the Internet of Things (loT), June 2023, Yokohama, Japan.

[i.10] S. Mante, R. Muppala, D. Niteesh, and A. M. Hussain: "Energy monitoring using LoRaWAN-based smart meters and oneM2M platform", in Proc. IEEE Sensors, October 2021.

[i.11] S. U. N. Goparaju, S. S. S. Vaddhiparthy, C. Pradeep, A. Vattem, and D. Gangadharan: "Design of an IoT system for machine learning calibrated TDS measurement in smart campus", in Proc. 7th IEEE World Forum Internet Things (WF-IoT), June 2021, pp. 877-882.

[i.12] oneM2M TS-0023 (V3.9.0): "Home Appliances Information Model and Mapping".

[i.13] oneM2M TS-0031 (V3.0.1): "Feature Catalogue".

[i.14] IEEE 802.11: "IEEE Standard for Information Technology--Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks--Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications".

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

oneM2M implementations: list of the implementations of the oneM2M standard

oneM2M numerosity objects: scalability of a oneM2M application

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 the oneM2M standard

real time requirements: timing constraints to be fulfilled by a system

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI TS 118 111 [i.6] and the following apply:

AC	Air Conditioning
ACP	Access Control Policy
ADN	Application Dedicated Node
AE	Application Entity
AE-CM	Application Entity - Crowd Monitoring
API	Application Program Interface
AQ	Air Quality
ASN-CSE	Application Service Node - Common Services Entity
ATCS	Adaptive Traffic Control System
C-DAC	The Centre for Development of Advanced Computing
CM	Crowd Monitoring

CO	Carbon monoxide
COAP	Constrained Application Protocol
CoSMiC	Common SMart iot Connectivity
COVID	Coronavirus Disease
CSE	Common Services Entity
CSF	Common Services Function
DEL	Data Exchange Layer
DML	Data Monitoring Layer
DPA	Data Platform Architecture
DSL	Data Storage Layer
ETSI	European Telecommunications Standards Institute
EV	Emergency Vehicle
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
HVAC	Heating Ventilation Air Conditioning
IEEE	Institute for Electrical and Electronic Engineers
IIIT	International Institute of Information Technology
IIIT-H	IIIT-Hyderabad
IN	Infrastructure Node
IN-CSE	Infrastructure Node - Common Services Entity
IoT	Internet of Things
IPE	Interworking Proxy application Entity
IUDX	Indian Urban Data Exchange
KPI	Key Performance Indexes
LORA	LOng RAnge (LoRa)
LoRaWAN	LoRa-Wide Area Network
LTE	Long Term Evolution
M2M	Machine-to-Machine
MN	Middle Node
MN-CSE	Middle Node - Common Services Entity
MQTT	Message Queuing Telemetry Transport
NoSQL	Not only Standard Query Language
OM2M	Open-Source platform for M2M communication (Eclips)
RAM	Random Access Memory
RDF	Resource Description Framework
SDT	Smart Device Template
SQL	Standard Query Language
TC	Technical Committee
TDS	Total Dissolved Solids
TR	Technical Report
TRAM	Traffic Monitoring and Management
TRAMMA	Traffic Monitoring and Management Application
TS	Technical Specification
TTF	Testing Task Force
URI	Uniform Resource Identifier
WF-IoT	World Forum Internet Things
Wi-Fi®	Wireless Fidelity (IEEE 802.11family)
WiSUN	Wireless Smart Ubiquitous Network

4 Method for Collecting Use Cases

The oneM2M template for the contribution of use cases [i.7] served as the source for structuring the clauses of the present document, which describes the use cases and the potential requirements to oneM2M derived from them. A call for input was issued to oneM2M delegates to provide real deployed use cases that may be examined as proof of concept examples for analysis. The following clauses capture the inputs received.

5 Use Case - Smart Campus

5.1 Description

This use case is based on the deployment made at International Institute of Information Technology, Hyderabad (IIIT). It is focused on smart campus with several domains involved such as: smart buildings, energy, water, streetlight, pollution, etc.

Localization: The campus of IIIT is in Telangana, India. IIIT is a residential institute spread over 66 acres.

The campus of IIIT University has deployed a Modular Data Platform Architecture (DPA) that is compliant with the Indian Urban Data Exchange - IUDX (https://iudx.org.in/) framework and oneM2M standards. The architecture consists of a oneM2M-based Data Monitoring Layer (DML) for seamless data accumulation from various sensor networks of a smart city such as air quality, water quality, and energy monitoring. This data is stored in the Data Storage Layer (DSL) using a multi-tenant architecture with multiple logical databases, enabling efficient and reliable data management. Finally, the Data Exchange Layer (DEL) enables secure data sharing in a format compliant with IUDX vocabulary.

5.2 Source

This oneM2M architecture is deployed at University Campus of International Institute of Information Technology, Hyderabad, India. Information has been collected based on [i.9].

5.3 Actors

M2M service provider: technical service and research department that provide the M2M service. This includes the server, sensors, networks and management of the oneM2M stack. The M2M service provider exposes API for the access of data.

External users: Based on authorization mechanism and specific layer, they can access the Data for campus management, research and industry activities.

5.4 Pre-conditions

The campus allows the deployment of sensors and connection to existing systems.

The M2M service provider proposes a oneM2M resources architecture, deploys the system and manages it.

5.5 Triggers

Some transitional phases will happen when the setting of the overall architecture and when new equipment should be connected (registration of new gateways on servers, registration of end devices on gateways).

On the permanent phase, the sensor data are sent to the IN-CSE through the oneM2M architecture. Through subscriptions and request response the external elements analyses and react to the data collected.

5.6 High Level Illustration

The proposed architecture consists of a DML, DSL and a DEL as illustrated in Figure 5.6-1. Multiple IoT nodes post data to the DML at predefined intervals depending on the parameters being monitored. The DML forwards this data using the subscription-notification CSF of oneM2M to the DSL, where it is parsed and ingested into a database. The data can be subsequently accessed by registered clients through the APIs defined in the DEL.

The architecture is deployed inside the campus with sensors, network, servers, etc. Several domains are covered: Smart spaces, Environment, Weather, Water, Energy, etc.

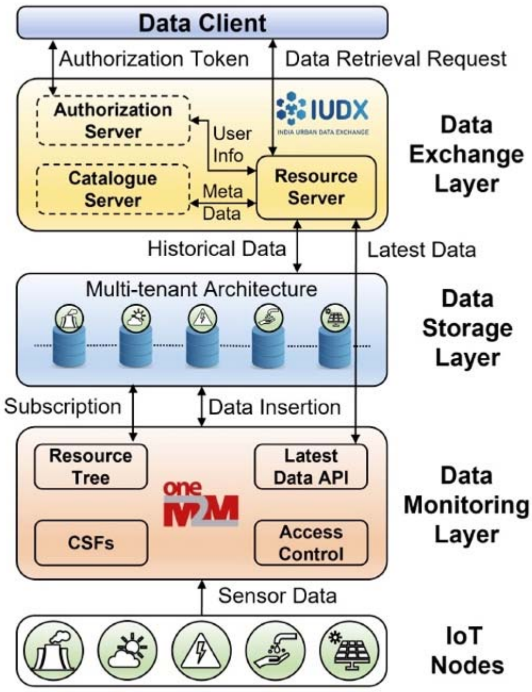

Figure 5.6-1: Solution deployed at IIIT-H campus

5.7 Normal Flow

Several types of equipment in different domains are connected to the oneM2M system:

a) Air Quality Monitoring: Several air quality parameters such as particulate matter (pm2.5 and pm10), temperature, relative humidity, and CO concentration are monitored through densely deployed sensor nodes to increase the spatial-temporal resolution of air quality data.

b) Crowd Monitoring: This sensor network monitors crowding of people to check the number of mask violations to avoid a COVID-outbreak inside the campus.

c) Energy Monitoring: Several energy parameters such as the individual phase currents and voltages, power factor, frequency, energy consumption, apparent and real power are monitored to understand the usage patterns and provide faster resolutions to power outages (see [i.10] S. Mante, R. Muppala, D. Niteesh, and A. M. Hussain, "Energy monitoring using LoRaWAN-based smart meters and oneM2M platform", in Proc. IEEE Sensors, Oct 2021, pp. 1-4.).

d) Water Quality Monitoring: Water quality parameters such as Total Dissolved Solids (TDS), and temperature are being monitored to avoid health problems caused by poor quality water, as detailed in [i.11] S. U. N. Goparaju, S. S. S. Vaddhiparthy, C. Pradeep, A. Vattem, and D. Gangadharan, "Design of an IoT system for machine learning calibrated TDS measurement in smart campus", in Proc. 7th IEEE World Forum Internet Things (WF-IoT), June 2021, pp. 877-882.

e) Smart Room Monitoring: Occupancy state and energy consumption of a room are monitored to adjust the air conditioning, lighting, and ventilation dynamically.

f) Solar Monitoring: Parameters such as energy generated in a day, signed active power, instantaneous frequency, output power factor, voltage, and current are monitored to efficiently analyse the solar energy generated by solar panels.

g) Weather Monitoring: Multiple weather monitoring stations are deployed to monitor parameters such as solar radiation, temperature, relative humidity, wind direction, wind speed, gust speed and dew point. This data is used for weather forecasting.

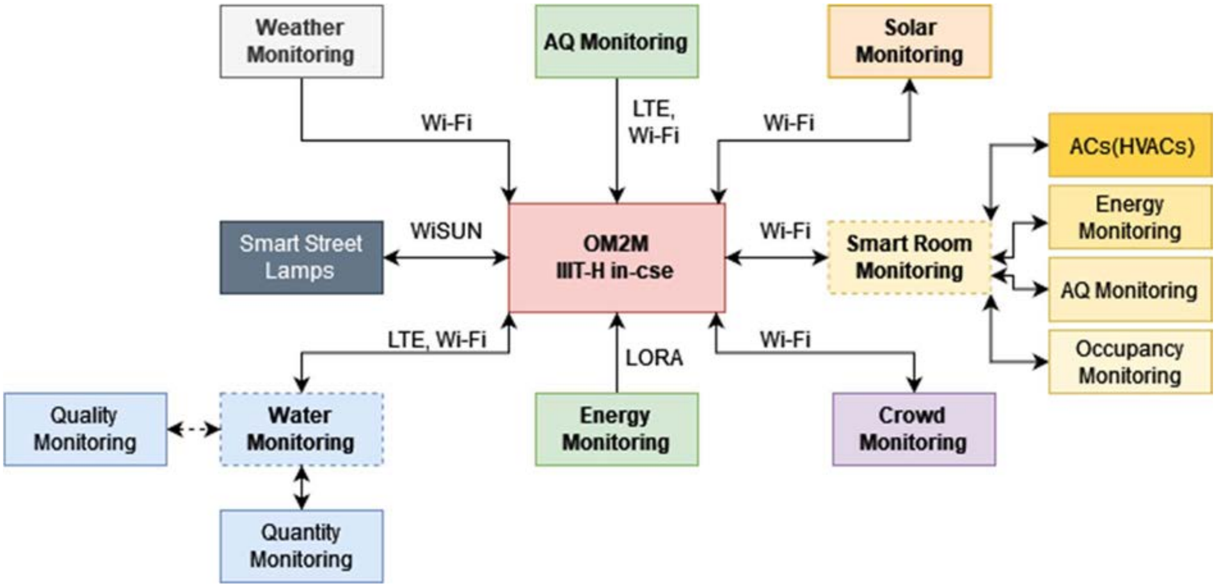

Figure 5.7-1: OM2M deployment at IIIT-H campus

The current deployment at IIIT-H uses one infrastructure node. As illustrated in Figure 5.7-1, different verticals use their respective communication protocols based on the interfaced sensors. The nodes use HTTP as the application protocol to publish periodic data (event triggered in the case of Occupancy) to their respective containers.

The former architecture adapted individual MN-CSE base for each respective vertical, which are interfaced to an IN-CSE. This former model was dropped since the deployments were implemented on the same base machine.

5.8 Alternate Flow

Not relevant.

5.9 Post-conditions

The data are stored in the specific database for long-term use.

5.10 oneM2M resources

Table 5.10-1 gives a detailed description of the effective exchange of message between the different parts of the architecture from sensors to gateways and servers.

Table 5.10-1: oneM2M exchanges of messages

Type of sensors	Number of equipment	Frequency of data sending for one equipment (Data instances/ publishing period)	Numbers of data in a message (Values per instance)	Communication protocol	Connected to gateway, server?
Air Quality Monitoring	11	4/min	12	LTE, Wi-Fi	IN-CSE
Crowd Monitoring	7	1/min	6	Wi-Fi	IN-CSE
Energy Monitoring	50	1/15 min	17	LORA	LORAWAN Gateway
Water Quality Monitoring	20	1/4hours	5	Wi-Fi	IN-CSE
Water Quantity Monitoring	25	1/min	5	LTE, Wi-Fi	External Restful APIs, IN-CSE
Smart Room AC Monitoring	91	1/min	27	Wi-Fi	Bacnet actuator, IN-CSE
Smart Room AQ Monitoring	18	1/min	7	Wi-Fi	IN-CSE
Smart Room Energy Monitoring	3	1/min	4	Wi-Fi	IN-CSE
Smart Room Occupancy Monitoring	7	Event triggered (delay of 1/min)	4	Wi-Fi	IN-CSE
Solar Monitoring	6	1/min	35	Wi-Fi	IN-CSE
Weather Monitoring	2	1/15 min	9	Wi-Fi	External Resful APIs, IN-CSE
Smart Street Lamps	30	1/min	8	WiSUN	WiSUN Gateway, IN-CSE

The data monitoring layer is based on oneM2M standards. It uses the inbuilt CSFs to accumulate data from various IoT nodes. For instance, every data point generated by a certain water quality node can be stored as a <contentInstance> resource inside the respective <container> resource. Multiple such containers can act as child resources to an Application Entity (<AE>) resource, which corresponds to the entire water quality sensor network.

Further, these <container> resources can be managed in groups by defining a <group> resource. Data forwarding to the DSL is enabled through <subscription> resources. Every non-ACP resource is linked to an <accessControlPolicy> (ACP) resource via an <accessControlPolicyID> (ACPI) attribute. ACPs govern the set of allowed operations that can be performed by a specific user in a given circumstance (time, IP address).

oneM2M also enables discovery and retrieval of resources with options to filter based on type, labels, and content size. Further, each <container> resource consists of two unique attributes ol, and la whose values act as Uniform Resource Identifiers (URIs) to retrieve the oldest and the latest <contentInstance> respectively. The DML is the first layer to receive any data point, preserving the highest degree of data freshness in the DPA. Hence, every latestData retrieval request received by the DEL exploits the la attribute provided by oneM2M.

The developed resource tree is illustrated in Figure 5.10-1. It consists of an IN-CSE, the root or parent for all the resources in the resource tree. IN stands for Infrastructure Node and CSE stands for Common Service Entity. The IN-CSE has multiple <AEs> and <cnts>, each <AE> corresponding to a sensor network, and each <cnt> to an IoT node. The <cnt> describes the node's data attributes and stores the incoming live data. Each sensor network has a unique <ACP> to enable controlled data inputs. For instance, in Figure 5.10-1, AE-CM is the application entity belonging to the Crowd Monitoring (CM) sensor network, which contains six nodes, CM-MG00-00 to CM-VN91-00. These nodes use the acp-crowd access control policy for publishing the data. Other sensor networks publish data to the corresponding AEs in a similar manner.

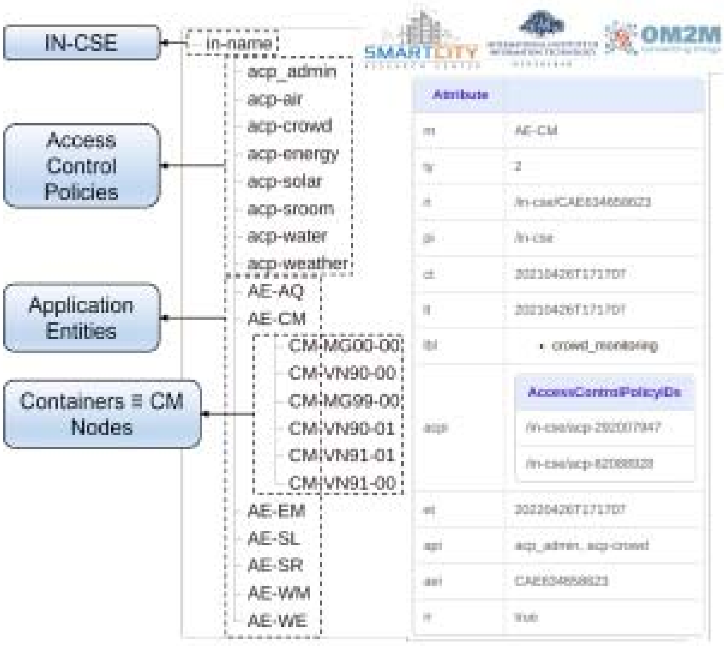

Figure 5.10-1: IIIT campus, resources tree

The current system has a dedicated container for each IoT node. Each such container has a Data Container which receives the data from the node. An individual subscription entity exists for each such data container.

A simple get request can also help in retrieving the data from a data container but, the probability of requesting same data point would drastically increase due variable frequency of the verticals which lead to adapt the subscription-based approach.

5.11 Potential Requirements

The experiments were conducted on a computer with an Intel® Core™ i5-8400 2.80 GHz processor, 8 GB of RAM, and a 64-bit Ubuntu 18.04 operating system with the opensource eclipse OM2M (https://eclipse.org/om2m). The primary purpose of this experiment was to mimic the DML's real-time data insertion and retrieval scenarios. A 12-hour performance test has been conducted on the DML, and the analysis revealed that it could handle 8 parallel users in the data-insertion scenario. As seen in Figure 5.11-1, the retrieval latency increased proportionally with the increasing parallel users, and DML was able to handle up to 600 parallel users with zero downtime in the data-retrieval scenario, while keeping latency well under 100 milliseconds.

The performance evaluation has been conducted in terms of insertions and retrieval latency, where latency is the number of successful requests served per minute. The testing revealed that 8 parallel nodes can create content instances in a container at any given instant, and around 600 parallel users can retrieve the latest content instance of a container from OM2M with an average latency less than 65 ms/request.

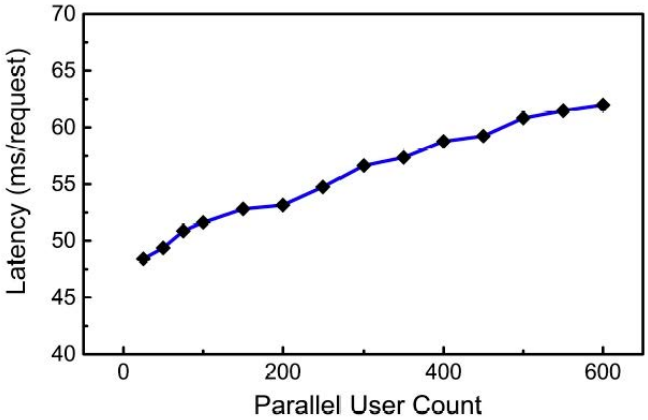

Figure 5.11-1: IIIT campus: OM2M performance

6 Use Case - Generalization of Event Trigger/Periodic Event IoT System

6.1 Description

This use case is a generic one based on event triggered computations and periodic computations on a full machine to machine IoT system. Concerning the deployed architecture, the number of end devices is parametrized as well as the number of gateways and servers.
Non-functional parameters characterize the behaviour of the system such as periodicity can be settled on the fly.

Non-functional constraints parameters related to response time, parameters on scalability are part of the description.

Such a scenario is typically encountered in intelligent building control systems with a multi-sensor environment, multi control-command on environmental actuators (heating, lighting, pollution management). This scenario is used to evaluate the scalability of the system with respect to the infrastructure.This scalability is measured through the criteria of real-time data recording capacity and their persistence.

6.2 Source

This use case is partially inspired by [i.1], clause 9.5 on Event Triggered Task Execution. The rest of the use case has been created from scratch.

6.3 Actors

- M2M Devices that produce periodical data.
- Gateway that integrates the oneM2M services of Middle nodes and process and react on certain values.
- Servers that integrate the services of Infrastructure nodes and process and react to certain values.

6.4 Pre-conditions

- Sensor Devices are configured to produce data periodically and send it to a Gateway device.
- Gateway Device is configured to work as the gateway for collecting data from some sensor device and generates requests to actuator devices.
- Actuator Devices are configured to accept request from a Gateway Device.

- Server Device is configured to work as an infrastructure node for collecting data from Gateway Devices and generate request either to the Gateway or the end devices.

6.5 Triggers

- **Transitional phase:**
  - Setting of the overall architecture: Instantiation of the IN_CSE on the server.
  - Registration of gateways on servers.
  - Registration of end devices on gateways.
- **Permanent phase:**
  - Production of data by the devices and registration on gateways.
  - Reaction of gateways according to sensors values.
  - Reaction of servers according to gateways values.

6.6 High Level Illustration

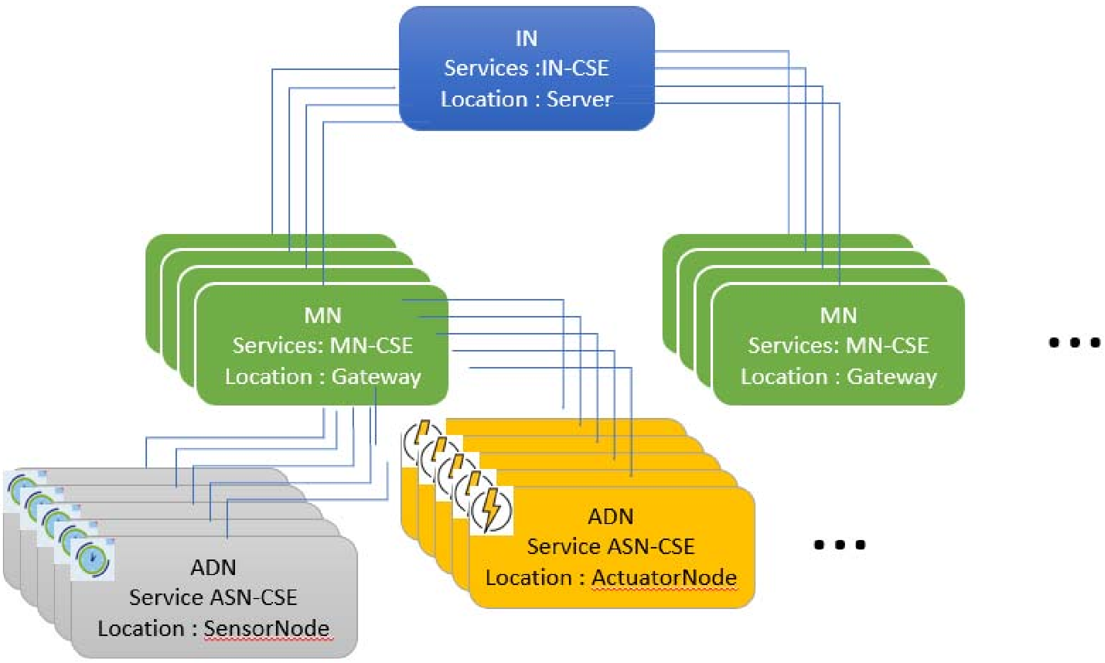

Figure 6.6-1: General view of a multi-level event-triggered and periodic infrastructure

6.7 Normal Flow

Normal Flow 1:

1) Sensor devices produce data periodically and register them on Gateway Devices.
2) Gateway device reports the collected data to Data Storage Server.
3) Server may request actuation on certain devices under certain constraints.

Normal flow 2:

1) Sensor devices produce data and register on the Gateway Device.

2) Gateway devices compute data.

3) Gateway may request actuation on certain devices under certain constraints.

6.8 Alternative flow

Alternative flow 1: Non Real-time registration on the Gateway or on the server

1) Sensor Devices produce data and register at a certain rate.

2) The registration time on the Gateway/Server is higher than the production rate of data leading to loss of samples.

Alternative flow 2: Non Real-time reaction of the Gateway or the server

1) Sensor Devices produce data and register at a certain rate.

2) The reaction time of the Gateway is higher that the production time of data leading to an inconsistent state of the system.

Alternative flow 3: Overflow on gateways related to *maxNrOfInstances* parameter or *maxByteSize* of containers

1) Sensor Devices produce data at a certain rate.

2) Detection of an overflow on a Gateway w.r.t data persistency (data storage) resulting in a loss by data overwriting.

Alternative flow 4: Overflow on gateways related to *Time Series Data Detecting* and *Reporting*

1) Sensor Devices produce data at a certain rate.

Data are missing on the timeseries resource producing a notification by the timeseries resource because the total number of missing data points becomes equal to or greater than the "minimum specified missing number of the Time Series Data" in *missingData* condition.

6.9 Post-conditions

Normal flow 1 & 2:

- Device data are periodically stored either on the gateway or the server.
- Actuation is produced on time according to the response time.

Alternative flow 1:

- Non-Real-time registration on the gateway or on the server.
- Data samples are missing on the gateway or on the server.

Alternative flow 2:

- Non-Real-time reaction of the gateway or the server.
- Reaction of the system is too low with respect to the environment dynamic.

Alternative flow 3:

- Overflow on gateways related to *maxNrOfInstances* parameter or *maxByteSize* of containers.
- Inconsistency between data production and data persistency on a database dealing to on a loss by data overwriting on the data queue.

Alternative flow 4: Overflow on gateways related to *Time Series Data Detecting and Reporting*

- A NOTIFY is reported because the total number of missing data points becomes equal to or greater than the "minimum specified missing number of the Time Series Data" in *missingData* condition.

6.10 Description of the deployed solution

Table 6.10-1: Deployed Solution

Type of - devices	Number of devices	Frequency of data sending for one equipment (Data instances/ publishing period)	Numbers of data in a message (Values per instance)	Communication protocol/transportation time	Connected to gateway, server?
Sensor_i	*Nb_sensor*	*Period/time_unit*	*Size_sensors*	*X hop TransTime*	*Gateway_gi*
Actuator j	*Nb_actuator*	*NA*	*Size_actuators*	*X hop TransTime*	*Gateway_Gi*
Gateway	*Nb_Gateway*	*Sporadique*		*X hop TransTime*	*Server_si*
Server	*Nb_server*	*Sporadique*		*X hop TransTime*	

6.11 Potential Requirements

- During the initialization phase

  Scalability: the system can absorb the registration requests with a variable ratio NbDevice/NbGateway and NbGateway/NbServer.

- During the permanent phase

  Reaction Time: the system is able to react in real-time to any change in the environment according to a *responsetime* prerequisite.

  Registration Time: the system can store on the fly the data produced by the environment i.e. the registration time for a data is less than the production time of this data.

  Persistency of Data: the system can implement data persistency according to specific data-based technologies (time series, NoSQL, etc.).

  Stressing the system: reaction time and registration time are checked in the context of a stressed system i.e. by increasing the ratio device/gateway and/or gateway/server.

7 Use Case - Traffic Light Control and Monitoring

7.1 Description

Traffic lights are deployed in all major cities to regulate the flow of traffic at intersections. A common problem that is experienced when deploying traffic lights is the determination of a schedule that will synchronize the flow of traffic through many intersections on a given route. One aspect of this synchronization problem is the inability to monitor and update the schedule of the traffic lights at these intersections. This represents a good use case for an IoT solution and a sub-component of a larger smart city deployment.

7.2 Source

This use case is partially inspired by [i.1], clause 8.1 Street Light Automation and clause 11 Transportation Use Cases.

The rest of the use case has been created from scratch.

7.3 Actors

M2M Service Provider: This is the organization that is hosting the oneM2M CSE service layer. This could be a commercial company or a government agency (such as a department within the city that manages traffic lights). The intent is that this service provider would host one or more oneM2M CSE instances such that all traffic lights within the scope of responsibility of the city agency would be represented in this CSE hierarchy.

Government Agency: This is the organization that is ultimately responsible for direct or indirect management and monitoring of the traffic lights at all intersections in the area of responsibility. This agency may sub-contract any aspects of this deployment to separate entities. The government agency represents the user that monitors and controls the operations of the traffic lights within the city.

Traffic Light: This is each individual traffic light at an intersection or any group of traffic lights that operate in the same exact manner (all lights are always the same within this grouping).

Traffic Light IPE: This is a oneM2M application entity specifically designed to interwork existing traffic lights to a oneM2M CSE and resource tree architecture.

Traffic Light Intersection: This is a grouping of individual traffic lights at a single intersection that are required to operate in a synchronized manner such that traffic flow is maintained, and accidents are prevented.

Traffic Light Intersection Chain: This is a grouping of separate traffic light intersections that are related by proximity to each other such as on a road where the traffic pattern expects vehicles to encounter multiple individual Traffic light intersections within the same journey.

7.4 Pre-conditions

The management and control of traffic lights is widely studied and there are multiple standards for specifying the control of traffic light timing. This use case assumes that a oneM2M resource tree structure is defined in a manner that generically monitors the states of the traffic lights and can use oneM2M device management features to specify control files used by traffic light controllers. This use case does not include real-time control of traffic lights such that the operation of the CSE would be in the critical path for the correct operation of the traffic control system:

1) A resource tree structure for each traffic light or traffic light grouping is defined.

2) A resource tree structure for a traffic light intersection is defined.

This use case assumes that the underlying network is designed to meet real-time requirements of the deployment. Meeting these real-time requirements is outside the scope of oneM2M and this study. Use of time sync and time series can be leveraged to achieve the accurate representation of the states of the system.

7.5 Triggers

There are no specific triggers since the traffic lights within a city are expected to be in operation at all times. There may be instances that require occasional commands that are transient in nature that are used to send signals that are intended to synchronize the traffic light intersections within a traffic light intersection chain.

7.6 High Level Illustration

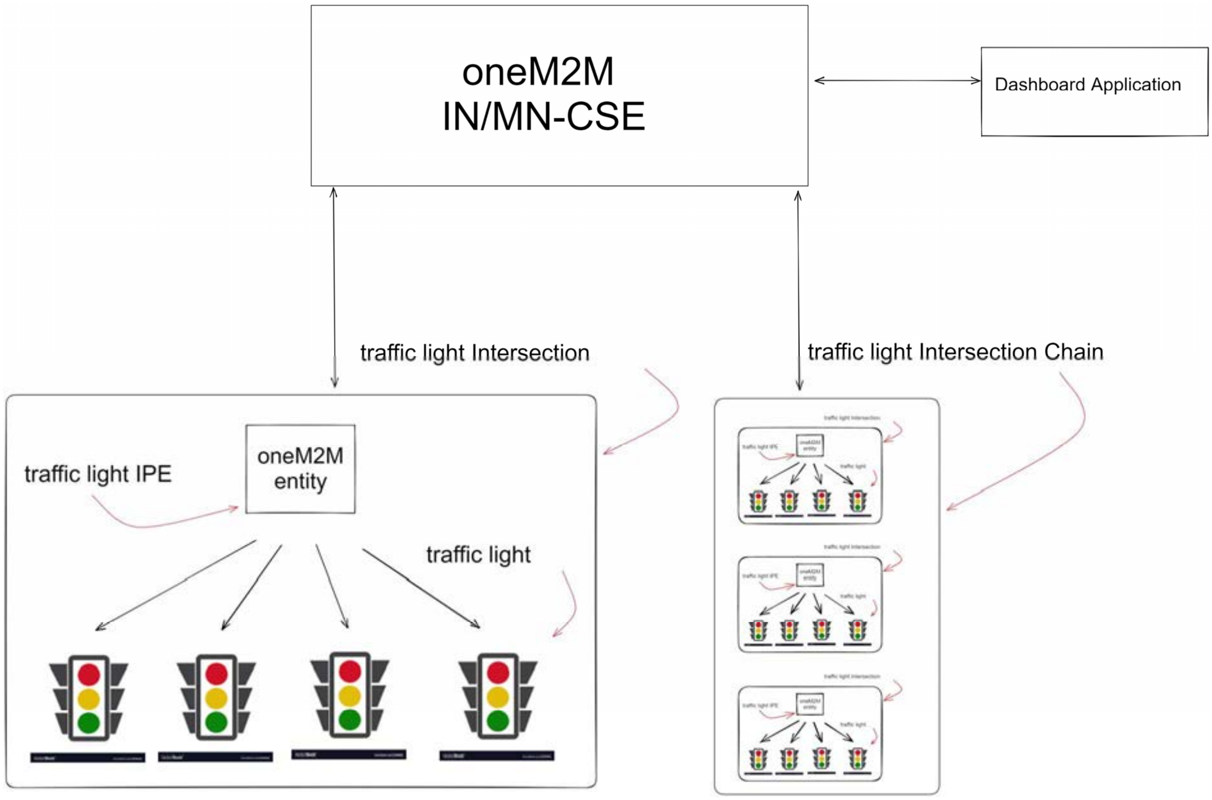

Figure 7.6-1: High Level Illustration

7.7 Normal Flow

The first call flow depicts the messaging that may exist for a single traffic light intersection. This call flow assumes that the individual traffic lights are not oneM2M entities and require a oneM2M AE Interworking Proxy Entity (IPE) to communicate with a oneM2M CSE. The specific oneM2M primitives sent by the IPE to the CSE are dependent on the resource tree structure that is defined. Therefore, as this study progresses there may be several variations shown and compared.

Another parameter that impacts the performance of this use case is the rate at which the model is updated. In one deployment option, the updates from the traffic light IPE can occur at fixed time intervals, such as one update per second. Alternatively, the updates can occur based on the change of states of all the lights at an intersection.

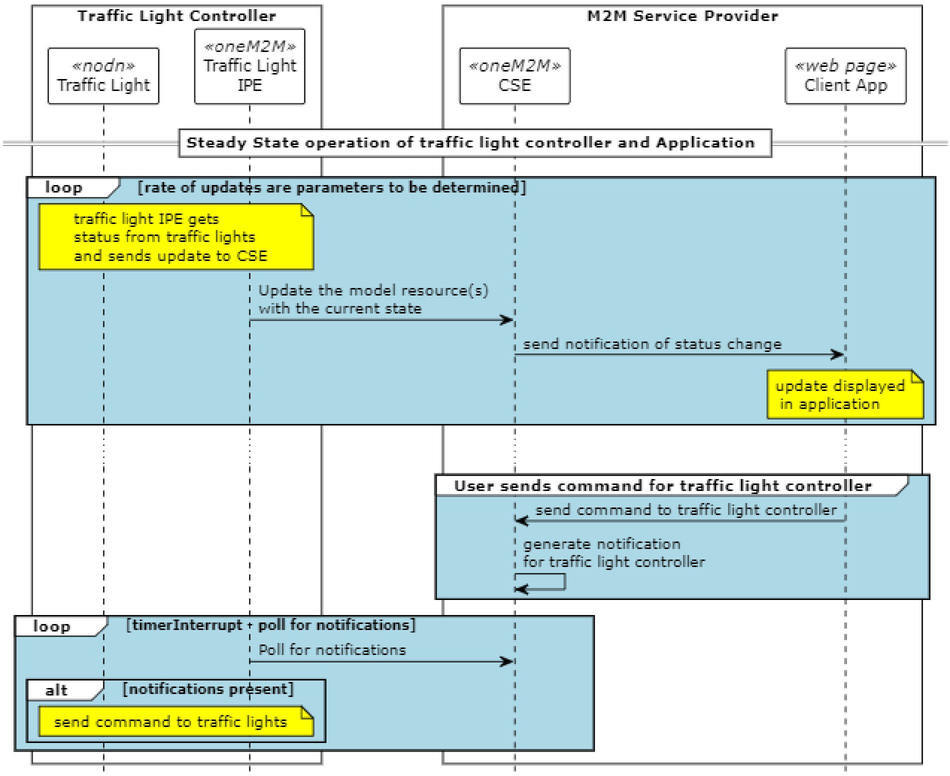

Figure 7.7-1: Call flow 1 - Traffic Light Intersection Chain

After consideration of the performance of the traffic lights at a single intersection it then becomes necessary to look at the performance of a larger Traffic Light Intersection Chain. This will entail understanding how many traffic light intersections can be supported by a single CSE implementation combined with the horizontal deployment of multiple CSE instances to manage larger numbers of intersections.

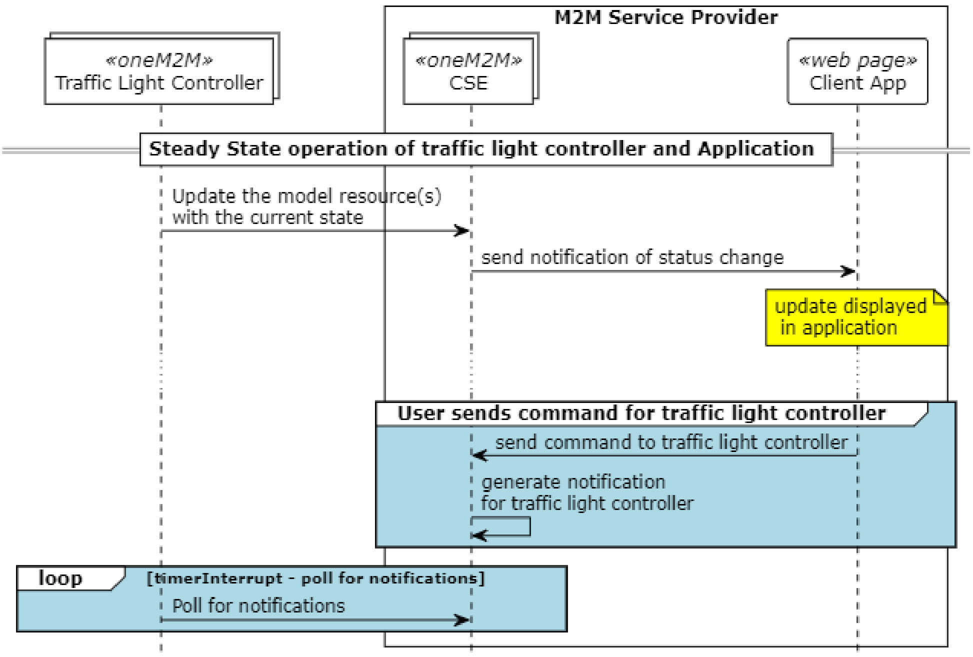

Figure 7.7-2: Call flow 2 - Traffic light intersection chain

7.8 Alternative flow

There are alternative flows that can be included, such as the impact of emergency vehicles that change traffic light patterns to facilitate safe passage through an intersection. However, these types of use cases are considered transient in nature and do not have a visible impact on the performance aspect of the system that is being considered for this study.

An alternate deployment hierarchy of oneM2M CSEs could include an IN-CSE that the controller application registers to and a series of MN-CSE deployments for collections of Traffic Light Intersection Chains. This can offer simplified application logic for the controller application and increased scalability options for the oneM2M Service Provider. This also presents opportunities to use additional features of oneM2M that can be used to implement this structure, such as retargeting or primitives and announcement of resources.

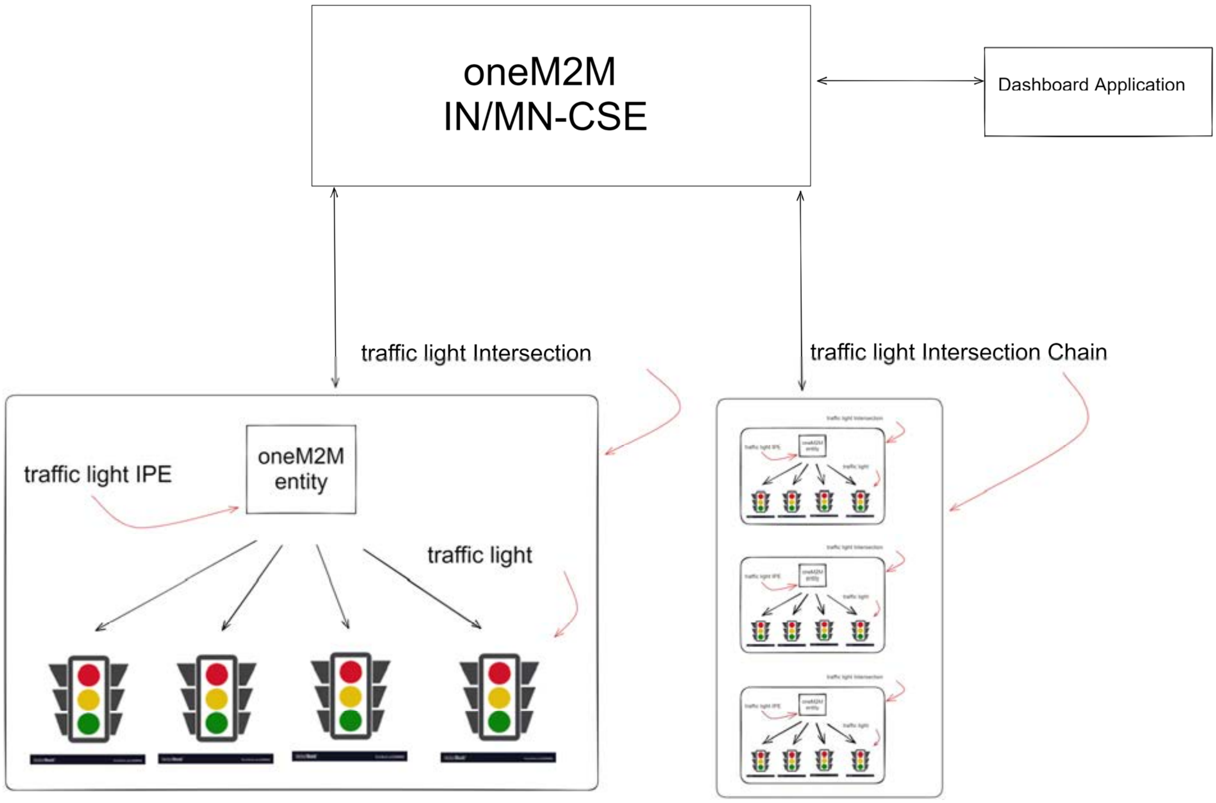

Figure 7.8-1: Alternative Flow

A second alternate flow, not shown, is increasing the number of applications that are getting status information from this system. Consider a mobile mapping application that shows traffic lights on a map, where the traffic light includes the current state of the light that applies to the current route (or direction of travel). This would entail issuing queries to locate the exact resources to target for primitives to get light status.

7.9 Post-conditions

None.

7.10 Description of the Deployed solution or oneM2M resources

7.10.0 Foreword

A variety of resource tree structures can be used to model these scenarios. There resource tree structures are described below along with the call flows associated with each variation. In the first resource tree structure a custom data model using <container> resources is defined. In the second resource tree structure a custom data model using <flexContainer> resources is defined. In the third resource tree structure a Smart Device Template based data model using <flexContainer> resources is used. The final resource tree structure is based on the actual deployment of a traffic light system.

7.10.1 Traffic Light model using custom <container> resource structure

When deploying an IoT system using oneM2M a simple approach to modelling is using oneM2M application entities, containers, content instances, and subscriptions resources to build a custom resource tree structure.

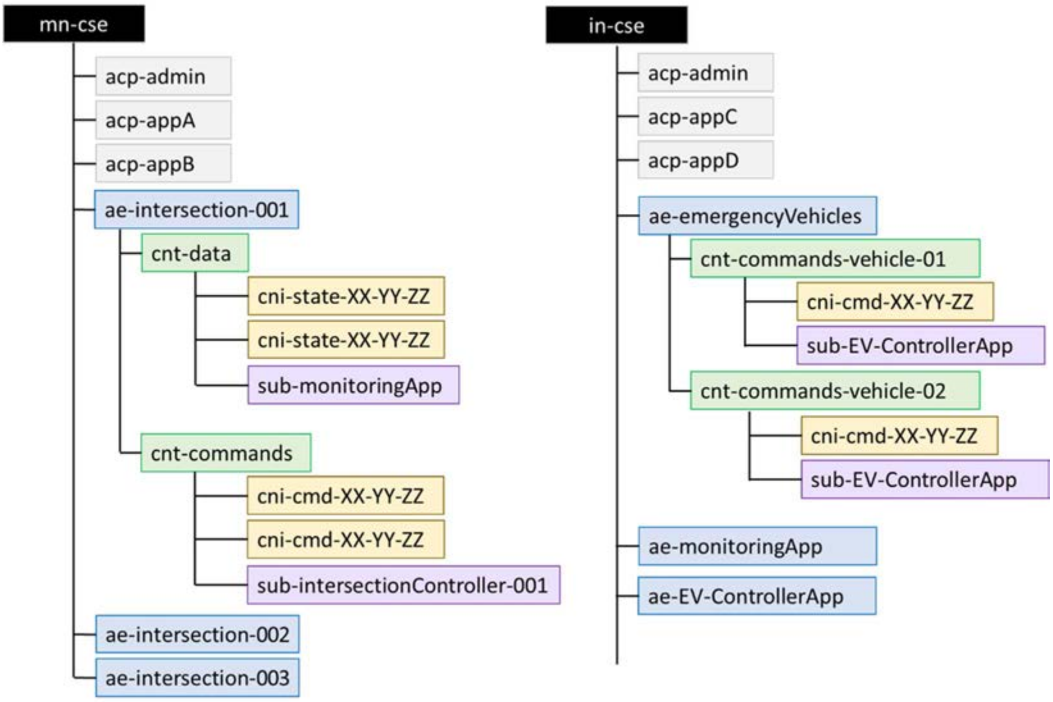

Figure 7.10.1-1: Traffic Light Model

In this scenario, each intersection is represented with a oneM2M application entity under a certain MN-CSE. This application entity will host two containers: data container and commands' container. In the first one, the intersection controlling application will store/report the state of the intersection whenever it changes. Monitoring applications can subscribe to this container to be notified by each new state of this intersection. In the second one, the application controlling the intersection creates a subscription to receive any new content instance (i.e. a command) sent by another application to effectively change the state of the intersection based on the "received" command.

On the IN-CSE, an application entity is created to represent emergency vehicles that can command the state of the intersection. Under this application entity, a container is created for each considered Emergency Vehicles (EVs). The container will store any "command" sent by the corresponding EV. As the EVs controlling application has subscribed to this container, this newly created command will notify the application to compute the set of the intersections to be commanded and send the necessary commands to the appropriate intersections.

Under the IN-CSE, there is an application entity for the monitoring application that receives any intersection state change (i.e. ae-monitoringApp), and another application entity that receives any EV command (i.e. ae-EV-ControllerApp).

Finally, on both CSE, oneM2M AccessControlPolicy resources are created to manage the access rights of the different applications.

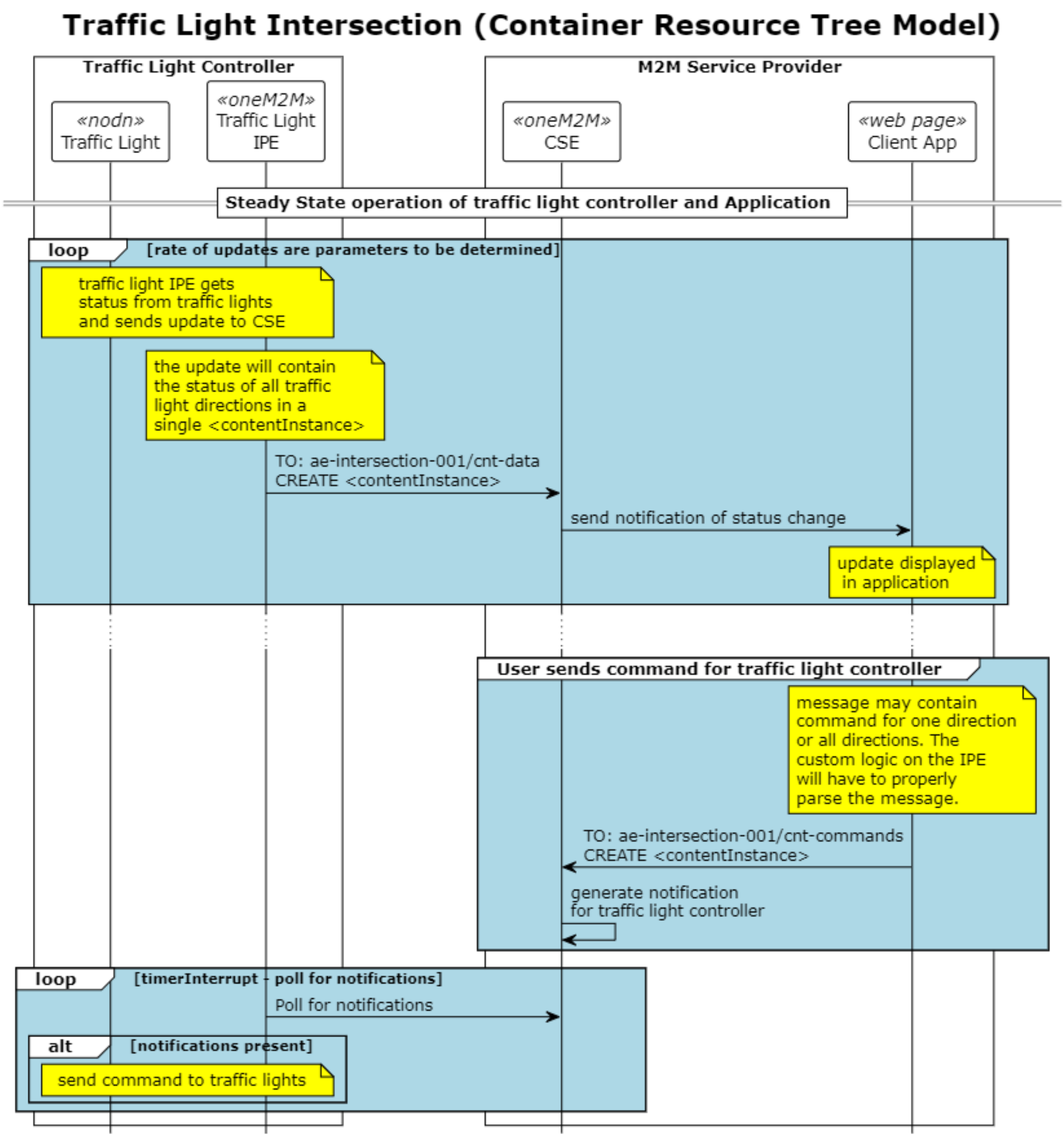

Figure 7.10.1-2: Traffic Light Intersection

The steady state message flow between the actors of this deployment includes the creating <contentInstance> resources by the IPE when an event occurs, such as a timer that indicates that the lights should transition to the next state. In this model the <contentInstance> will contain a custom data model that is opaque to the oneM2M CSE. The data model can include anything, but for this study the content will include the state of all the traffic lights at the intersection. This means that there will be one <contentInstance> create resource for every state change. When the <contentInstance> is received by the CSE, a notification is sent to each entity that has subscribed to this resource. This is generically modelled as a dashboard application.

Likewise, when a control signal is sent to the traffic light, such as may be done when an emergency vehicle is passing through the intersection, a <contentInstance> is created that commands all the lights at the intersection (or any desirable subset).

7.10.2 Traffic Light model using custom <flexContainer> resource structure

When deploying an IoT system using oneM2M another approach to modelling is through the use of oneM2M application entities, flexContainers, and subscriptions resources to build a custom resource tree structure.

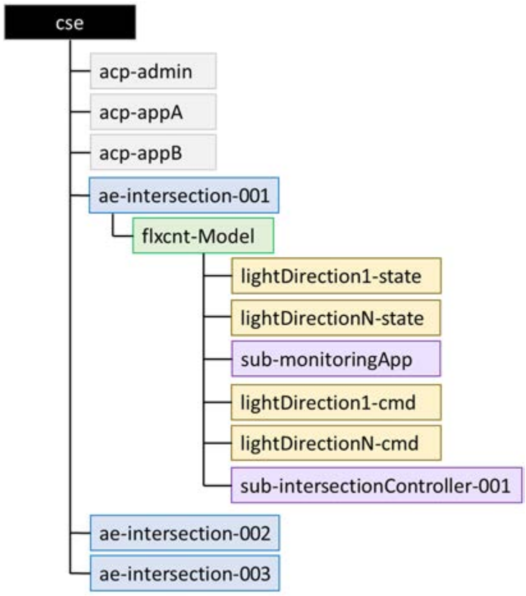

Figure 7.10.2-1: Traffic Light Model

In this scenario, just like the previous deployment, each intersection is represented with a oneM2M application entity under a certain MN-CSE. This application entity will host a single flexContainer that has attributes representing the state of each traffic light in the intersection and the attributes to send commands to the traffic light controller. The intersection controlling application will update attributes of the flexcontainer resource when the state of any of the lights in the intersection changes. As in the previous deployment, monitoring applications can subscribe to this flexContainer to be notified by each new state of this intersection. Also, the application controlling the intersection creates a subscription to receive any updates to attributes used for sending commands to effectively change the state of the intersection based on the "received" command.

On the IN-CSE, the application structure remains the same as the previous scenario. Finally, on both CSE, oneM2M AccessControlPolicy resources are created to manage the access rights of the different applications.

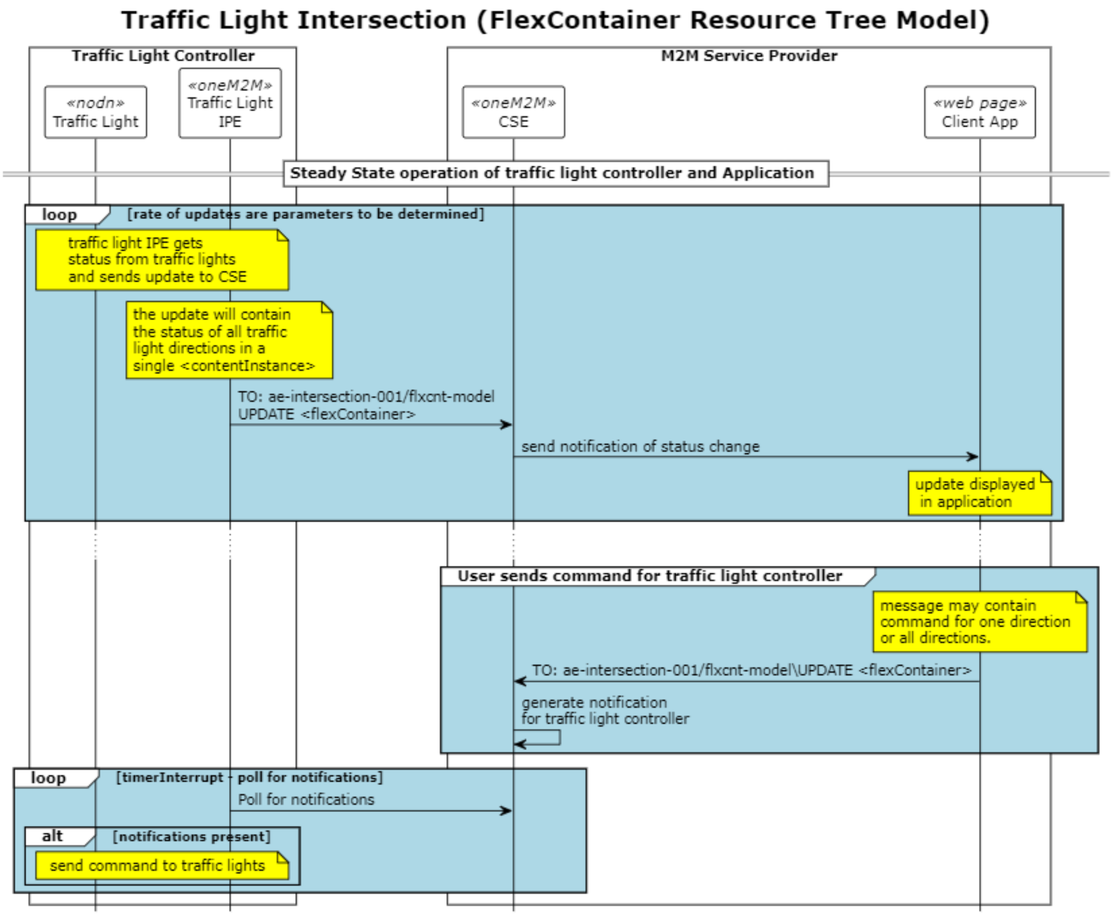

Figure 7.10.2-2: Traffic Light Intersection

The steady state message flow between the actors of this deployment includes updating <flexContainer> resources by the IPE when an event occurs, such as a timer that indicates that the lights should transition to the next state. In this model the <flexcontainer> will contain a state attribute for lights facing each direction in the intersection. The data in attributes of a <flexcontainer> are not opaque to the oneM2M CSE. Therefore, allowing for advance use of filter criteria to get desired data from the CSE. There will be one <flexcontainer> update primitive for every state change. When the update <flexcontainer> primitive is received by the CSE, a notification is sent to each entity that has subscribed to this resource. This is generically modelled as a dashboard application.

Likewise, when a control signal is sent to the traffic light, such as may be done when an emergency vehicle is passing through the intersection, the <flexContainer> is updated with the attributes that command any of the lights at the intersection (or any desirable subset).

7.10.3 Traffic Light model using Smart Device Template resource structure

When deploying an IoT system using oneM2M there is an approach to modelling that can be standardized by using oneM2M Smart Device Template resource tree structures defined in oneM2M TS-0023 [i.12]. The SDT resource tree structure also uses oneM2M application entities, flexContainers, and subscriptions resources to build a standard resource tree structure for the type of device being modelled.

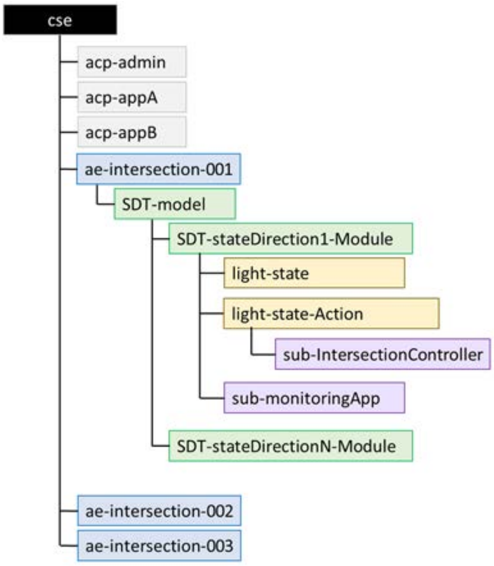

Figure 7.10.3-1: Traffic Light Model

In this scenario, just like the previous deployments, each intersection is represented with a oneM2M application entity under a certain MN-CSE. This application entity will host a top level flexContainer resource that represents a traffic intersection (note that this model does not exist in oneM2M at this time as is used for illustration purposes). There is a child module implemented with <flexContainer> resources to represent a single traffic light direction. The module that represents traffic lights in a specific direction have an attribute that indicates the state of the light in that direction and another child <flexcontainer> that is used for changing the state of that light for that direction. Using the SDT model there are many more resources needed to represent the same information as the custom models described above. However, using the SDT model provides a standardized approach to modelling all devices.

As in the previous deployment, monitoring applications can subscribe to these flexContainer resources to be notified by each new state of this intersection. Also, the application controlling the intersection creates a subscription to receive any updates to attributes used for sending commands to effectively change the state of the intersection based on the "received" command.

On the IN-CSE, the application structure remains the same as the previous scenario. Finally, on both CSE, oneM2M AccessControlPolicy resources are created to manage the access rights of the different applications.

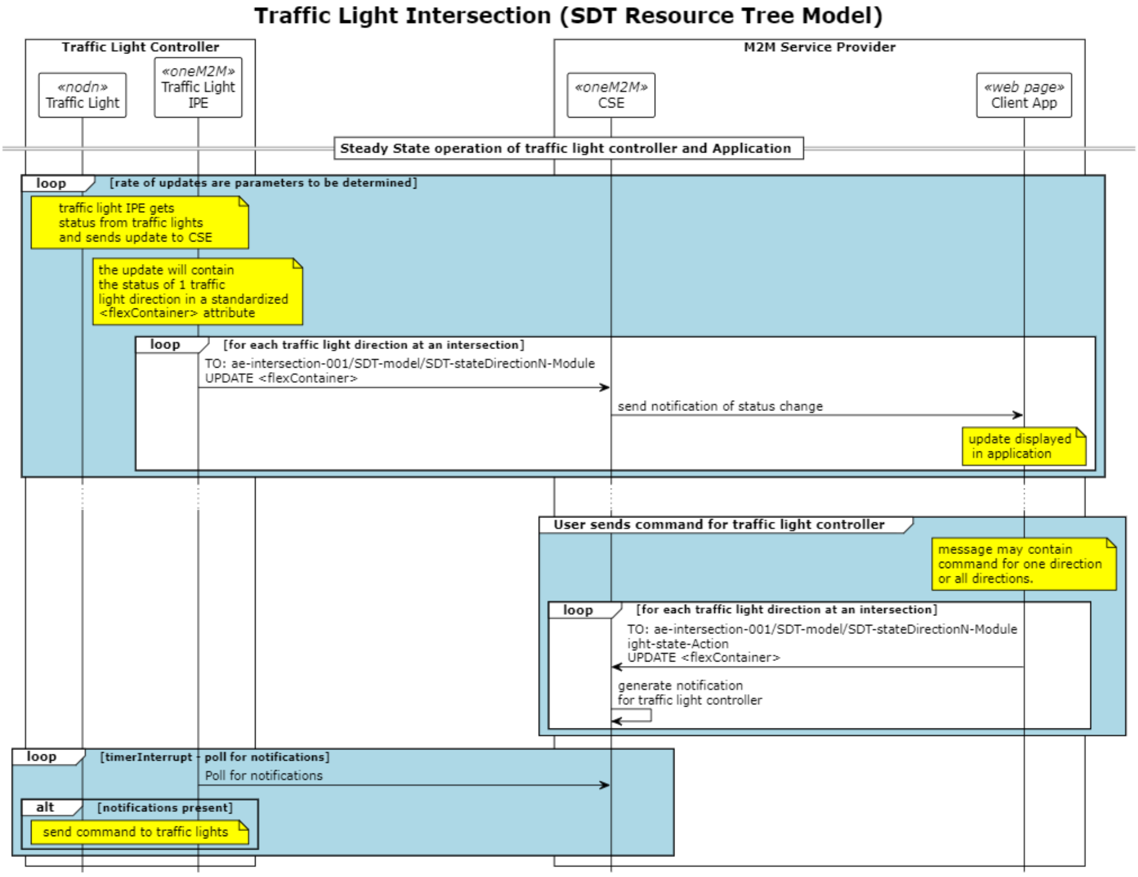

Figure 7.10.3-2: Traffic Light Intersection

The steady state message flow between the actors of this deployment includes updating <flexContainer> resources by the IPE when an event occurs, such as a timer that indicates that the lights should transition to the next state. In this model each <flexcontainer> module representing a single direction in the intersection will need to be updated. The data in attributes of a <flexcontainer> are not opaque to the oneM2M CSE. There will be multiple <flexcontainer> update primitives for every state change, one update for each light that changes at the same time. When each update <flexcontainer> primitive is received by the CSE, notifications are sent to each entity that has subscribed to these resources. This is generically modelled as a dashboard application.

Likewise, when a control signal is sent to the traffic light, such as may be done when an emergency vehicle is passing through the intersection, multiple <flexContainer> resources are updated with a single attribute defined by the action model resource.

7.10.4 Traffic Light model using C-DAC implementation resource structure

Indian organization, C-DAC, has implemented a traffic controller using their oneM2M implementation known as Common SMart iot Connectivity (CoSMiC). There are two types of applications in this deployment, C-DAC Traffic controller and Traffic Monitoring and Management Application (TRAMMA). This is an actual deployment using the custom container approach, described in clause 7.10.1, with realized deployment constraints.

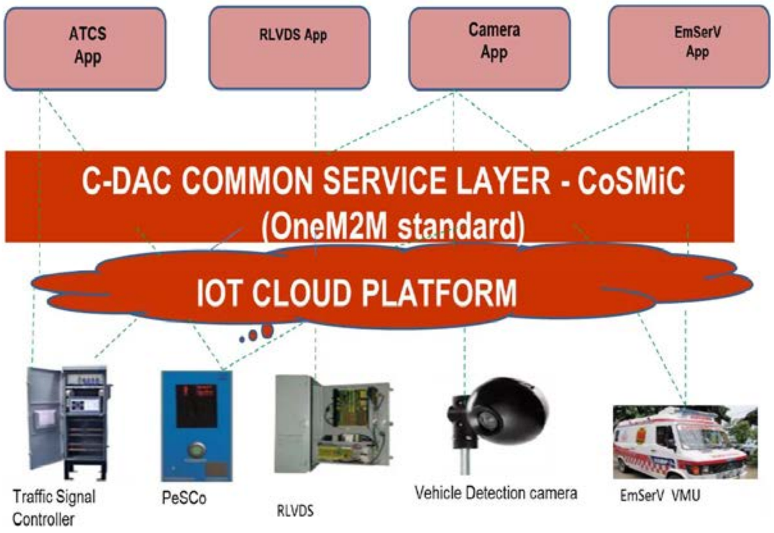

Figure 7.10.4-1: C-DAC COMMON SERVICE LAYER

Figure 7.10.4-1 shows the deployment architecture including several services. The present document is focused on the Traffic Signal Controller and the ATCS Application.

The resource tree structure for the traffic light deployments uses an AE for each intersection where the traffic light hardware registers upon startup and creates two containers, "CONTAINER" used for data from the traffic light hardware and "DESCRIPTOR" used for metadata about the intersection.

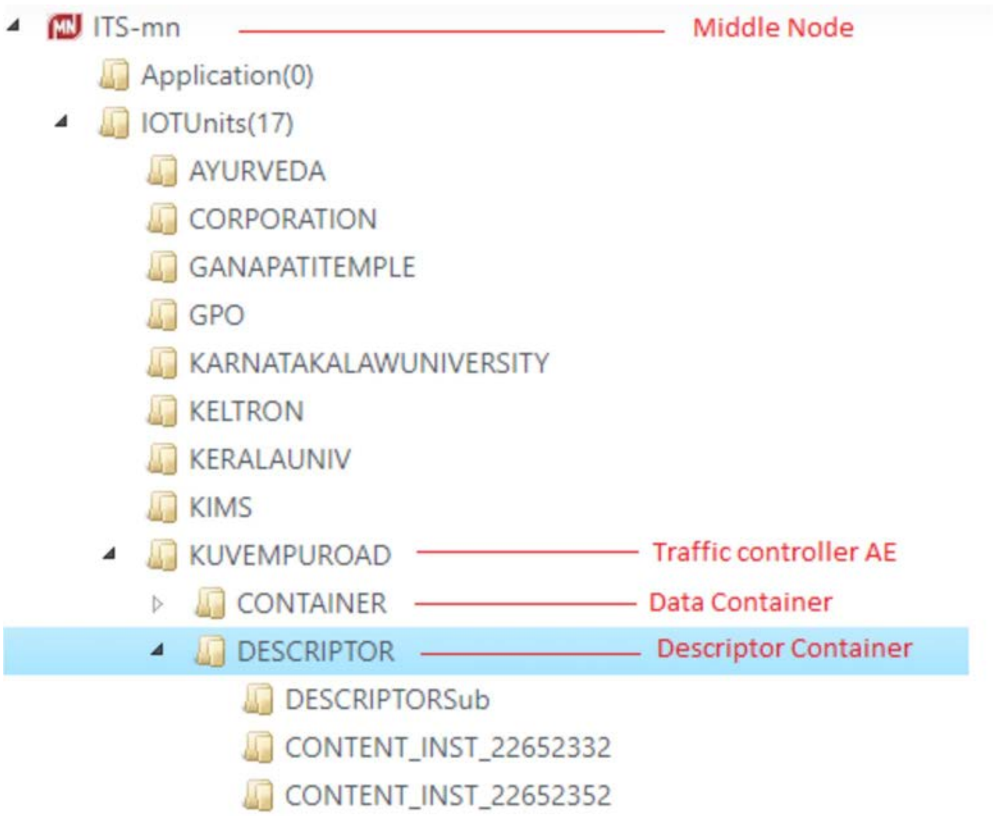

Figure 7.10.4-2: The resource tree structure for the traffic light

The "DESCRIPTOR" resource is used by a management console application that monitors and controls the intersection. Labels are used by the management console application to discover resources containing traffic lights and information in the <contentInstance> resources under the DESCRIPTOR <container> are used in the application. An example on the data in the <contentInstance> is:


```
{
"LAT": 8.510346,
"LONG": 76.960576,
" PRODUCT":"TRAFFICCONTROLLER/TRAMM,
"DEVICETYPE": ADN/IPE,
"PROTOCOL": HTTP/MQTT/COAP,
"ID":6
}
```


The "CONTAINER" resource contains data from the traffic controller hardware. The traffic controller sends all data messages to this <container> resource, which is subscribed to by the Traffic Monitoring and Management Application, using a custom data scheme for each message type. The <contentInstance> data structure contains a packet ID value that is used to decode the payload of the message. Table 7.10.4-1 shows the types of data packets and the frequency of each message.

Table 7.10.4-1: Container Content Instance Frequencies

Container Content Instance		
Packet ID	**Packet Info**	**Frequency**
1	Traffic Signal controller lamp change packet	Variable (3 to 99) seconds
2	Traffic Signal controller lamp change packet	5 seconds
14	Transit signal priority Packet	Not frequently
15	Red light violation feature packet	60 seconds
5	Event packet (Instantaneous)	Variable (5 to 900)
5	Event packet (on Request)	Received from controller once on request
36	Traffic Signal controller Utility Packet	Received from controller once on request
8	Traffic Plan Download Acknowledgement packet	Received from controller once on as acknowledgement
9	Traffic Plan Upload packet	Received from controller once on request
29	Traffic Signal controller Amber lamp switching Packet	Variable (3 to 99) seconds
30	Traffic signal Day Start Predict Packet	Variable 1 to 24 per day

In a similar manner, the Traffic Monitoring and Management Application places some control data in a <container> resource for the traffic controller hardware. The Traffic Monitoring and Management Application sends data messages to this <container> resource, which is subscribed to by a corresponding traffic controller, using a custom data schema for each message type. The <contentInstance> data structure contains a packet ID value that is used to decode the payload of the message. Table 7.10.4-2 shows the types of data packets and the frequency of each message.

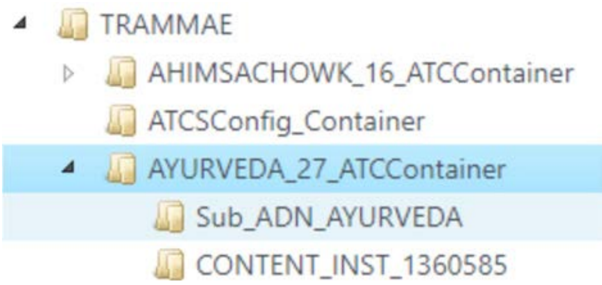

Figure 7.10.4-3: TRAMMA Resource Tree StructureTraffic Monitoring and Management Application

Table 7.10.4-2: TRAMMA Container Content Instance Frequencies

Container Content Instance		
Packet ID	**Packet Info**	**Frequency**
1	Adaptive Server Control packet	3 to 200 seconds
2	Remotely Traffic Signal controller ATCS Mode Change packet	Once on user intervention
4/5	Remotely Traffic Signal controller Time Update packet	Once per day
6	Event Request Packet	Once on day or on user intervention
7	Traffic Signal controller Utility Request packet	Once on user intervention
8	Traffic Signal Plan Download packet	Once on user intervention
9	Traffic Signal Plan Upload Request Packet	Once on user intervention

The following assumptions are made for the communication protocols:

1) Cellular IP communication from the traffic light IPE to the oneM2M CSE hosted in a private/public cloud service. Prefer CoAP over cellular.

2) Wi-Fi or Ethernet communications for the controller application using HTTPS.

3) Cellular IP communications for alternate scenario for mapping displays using HTTPS.

7.11 Potential Requirements

1) Show how many traffic light intersections can be hosted by a single CSE (or performance of a single CSE and the number of traffic light intersections hosted is increased).

2) Show how many primitives are used for different deployment architectures (primitive rates), per CSE and control application(s).

3) Show how many mobile applications can be supported (this may require deploying a CSE to support this application where limited information is "announced" about a traffic light).

4) Identify deployment challenges, such as increasing delays when a CSE becomes "loaded".

8 Conclusion

The use cases described in clauses 5, 6 and 7 evaluate several concrete cases of oneM2M deployments in different IoT domains. The goal is to have deployment scenarios that cover as many of the oneM2M features as possible. Table 8-1 shows the high-level features from oneM2M TS-0031 [i.13] that are covered by the described uses. Additional use cases may be defined in a later update to the present document to increase the coverage of these features and ensure that there is a good analysis of the scalability of deployments.

Table 8-1: oneM2M TS-0031 [i.13] Features Coverage

Feature from oneM2M TS-0031	Covered (Yes or No)
Binding and Serialization	Yes
Resource Identification	Yes
Request Handling	Yes
CRUD of common and universal attributes	Yes
CSE registration	Yes
AE registration	Yes
Configuration of container for data sharing	Yes
Managing content instances	Yes
Managing flexcontainers	Yes
Configuration of time series	Yes
Managing time series instances	Yes
Resource discovery	Yes
Configuration of group	No
Fan out using group	No
Configuration of subscription for data Notification	Yes
Trigger notification pertaining to subscription	Yes
Configuration of access control policy	Yes
Announce a Resource	No
Announced Resource Procedures	No
Configure a Polling Channel	No
Long Polling Procedures	No
Configure an Event Based Statistics Collection	No
Service Statistics Collection Procedures	No
Semantic annotation	No
Semantic discovery	No
Modelling of HAIM devices	Yes
Device Triggering	No

Annex A: Change History

Date	Version	Information about changes
January 2023	0.0.1	Bootstrapping Intro and Table of Contents of the document
January 2023	0.0.2	Typos founded during the meeting
February 2023	0.0.3	Add clause 5 from T. Monteil Add clause 6 from M.A. Peraldi-Frati (INRIA)
February 2023	0.0.4	Add clause 7 from B. Flynn (Exacta GSS)
March 2023	0.0.5	Implementing issues and suggestions raised on March 6 TTF Meeting
March 2023	0.0.6	Typos by L. Liquori (INRIA)
March 2023	0.0.7	Modifications& comments from M.A. Peraldi-Frati (INRIA)
March 2023	0.1.0	Final Draft version for review at TC SmartM2M #65
March 2023	0.1.1rc1	Remove personal information and contact details
March 2023	0.1.1rc2	Formatting, proof reading by L. Liquori (INRIA)
July 2023	1.1.1	Verification by ETSI Technical Officer for Publication pre-processing with editHelp!

History

Document history		
V1.1.1	September 2023	Publication