ETSI TR 103 674 V1.1.1 (2021-02)

SmartM2M;
Artificial Intelligence and the oneM2M architecture

Reference
DTR/SmartM2M-103674

Keywords
architecture, artificial intelligence, IoT, oneM2M

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 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

List of figures

List of tables

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.

Introduction

The resurgence of Artificial Intelligence (AI) is largely due to its ability to use techniques coming from both Big Data and Machine Learning (ML) and to demonstrate a vast range of promising new services (often still at prototype level). Artificial Intelligence technologies are applied in a large range of Industries with the expectation to provide added value to most ICT systems, including IoT systems. In its Communication COM(2018) 237 [i.9] on "*Artificial Intelligence for Europe*", the European Commission notes that "*Like the steam engine or electricity in the past, AI is transforming our world, our society and our industry*" and intends to "*facilitate access of all potential users, especially small and medium-sized enterprises, companies from non-tech sectors and public administrations, to the latest technologies and encourage them to test AI*".

The most spectacular achievements of AI are largely associated to the deployment of AI algorithms within existing systems supported by very powerful data management and computing capabilities. In the target systems (including IoT systems), AI is gradually integrated within various kinds of networks and supports a growing number of deployment models, in particular the cloud-based ones. To achieve excellence, AI applications need to be supported by a proper architectural structure, a capacity to easily integrate technology building blocks and components and the need to guaranty continued interoperability.

oneM2M, the global standards Partnership Project for M2M communications and the IoT, has published a very large set of oneM2M specifications (e.g. oneM2M TS-0001 [i.2]), designed not only to enable basic connectivity between applications and devices, but also to offer a broader support to higher levels of interoperability (such as semantic interoperability) and to security by enabling end-to-end secure information exchange between any devices or servers, as well as implementing dynamic access control. The oneM2M specifications are constantly evolving in order to take into account new advances in technology and Artificial Intelligence is one of them.

In order to maximize the benefits of integrating Artificial Intelligence and Machine Learning (ML), oneM2M has to, on the one hand, support the data-centric approach of AI/ML and its huge requirements in terms of resources available in the cloud domain as well as at the edge of the IoT network. On the other hand, AI is also an opportunity for oneM2M to provide open solutions to applications and services developers together with maintaining and enlarging its core asset of support to interoperability. The present document analyses the implications of AI on IoT systems and, as first priority, the oneM2M architecture.

The present document is complemented by the ETSI TR 103 675 [i.1] which addresses the development of a Proof-of-Concept based on a Use Case analysed and selected in the present document.

1 Scope

The present document is addressing the issues related to the introduction of AI into IoT systems and, as first priority, into the oneM2M architecture.

The following points are discussed:

- Identification of Relevant use case related to the introduction of AI in IoT systems.
- Analysis of the main implications of this use case to the oneM2M architecture.
- The selection of a relevant use case in view of its implementation as a Proof-of-Concept.

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 103 675 (V1.1.1): "SmartM2M; AI for IoT: A Proof of Concept".

[i.2] oneM2M TS-0001 (V4.8.0): "oneM2M Functional Architecture Baseline".

[i.3] StandICT, 2019: "ICT Standards and Ongoing Work at International Level In the AI Field - A Landscape Analysis".

[i.4] ISO/IEC CD TR 24030: "Information Technology - Artificial Intelligence (AI) - Use Cases".

[i.5] oneM2M TS-0034 (2019): "Semantic Support".

[i.6] ETSI TS 103 264: "SmartM2M; Smart Applications; Reference Ontology and oneM2M Mapping".

[i.7] ETSI GS CIM 009: "Context Information Management (CIM); NGSI-LD API".

[i.8] ISO/IEC 2382-31: "Information technology -- Vocabulary -- Part 31: Artificial intelligence -- Machine learning" (withdrawn).

[i.9] Communication COM(2018) 237: "Communication from the Commission to the European Parliament, the European Council, the Council, the European Economic And Social Committee and the Committee of the Regions".

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

artificial intelligence: system's ability to correctly interpret external data, to learn from such data, and to use those learnings to achieve specific goals and tasks through flexible adaptation

machine learning:

- process by which a functional unit improves its performance by acquiring new knowledge or skills, or by reorganizing existing knowledge or skills; or
- scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead.

NOTE 1: The first definition in this choice of two is from ISO/IEC 2382-31:1997 [i.8] and this standard has been withdrawn.

NOTE 2: It is seen as a subset of artificial intelligence.

oneM2M: Partnership Project (EPP) on M2M launched by a number of SSOs including ETSI

Standards Development Organization (SDO): standards setting organization that has a formal recognition by international treaties, regulation, etc.

NOTE: In the present document, SSO is used equally for both Standards Setting Organization or Standards Development Organizations (SDOs).

Standards Setting Organization (SSO): any entity whose primary activities are developing, coordinating, promulgating, revising, amending, reissuing, interpreting or otherwise maintaining standards that address the interests of a wide base of users outside the standards development organization

3.2 Symbols

Void.

3.3 Abbreviations

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

AI	Artificial Intelligence
API	Application Programming Interface
ICT	Information and Communication Technology
IoT	Internet of Things
ML	Machine Learning
PoC	Proof-of-Concept
SDO	Standards Development Organization
SSO	Standards Setting Organization
TR	Technical Report

4 Artificial Intelligence for IoT

4.1 The place of AI in IoT Standardization

The introduction of Artificial Intelligence in IoT systems is a very broad topic analysed (and possibly designed and implemented in prototypes or real systems) from a variety of angles. However, the scope of the present document can be clarified by the following observations:

- In the remainder of the present document, the term AI should be considered as AI/ML.
- The present document addresses specifically the place of AI in IoT Standardization and the impact of AI on the oneM2M architecture in particular.

Initial studies on the topic of AI standardization have been initiated by several Standardization Organizations (SDOs and SSOs). The landscaping done by the StandICT project in 2019 (see [i.3]) has identified 5 SDOs/SSOs amongst which four are already active in AI standardization (namely IEEE, ISO/IEC, ITU-T and ETSI) and one about to launch activities (namely CEN/CENELEC). This picture is constantly evolving as new Work Groups, Focus Groups, etc., are being created in active organizations, while new organizations join the AI standardization community. However, many of these activities address the global, AI landscape of AI and people-centric issues related to ethics and trustworthiness. The present document focuses on the specific aspect of AI for IoT, where the population of connected devices, relative to people, is projected to be several orders of magnitude greater.

The present document does not intend to address the overall AI in IoT standardization landscape but to focus on two major specific aspects. The first one deals with the definition of a broad set of representative use cases that are relevant to the introduction of AI in IoT, i.e. that address the most pressing issues (e.g. the data related ones). The other point of focus is the analysis of the impact of AI on IoT architectures, in particular the oneM2M service layer. This dual angle of analysis provides the foundation to select candidate use case for implementation and validation in the associated Proof-of-Concept phase of work (undertaken in ETSI TR 103 675 [i.1]).

4.2 Artificial Intelligence and IoT architectures

The flow of data has an important bearing on architectural components for IoT solutions. The typical focus is on data that leads to some form of decision being taken and is classified as 'user-plane' data. The basic model for IoT solutions begins with sourcing data from IoT devices (illustrated in left-hand side of Figure 1). This data then passes through a signal processing and machine learning process to extract key features and to represent them as knowledge-based objects. The next stage of processing involves the application of rules-based AI in areas related to reasoning, decision making, supervision and explainable AI.

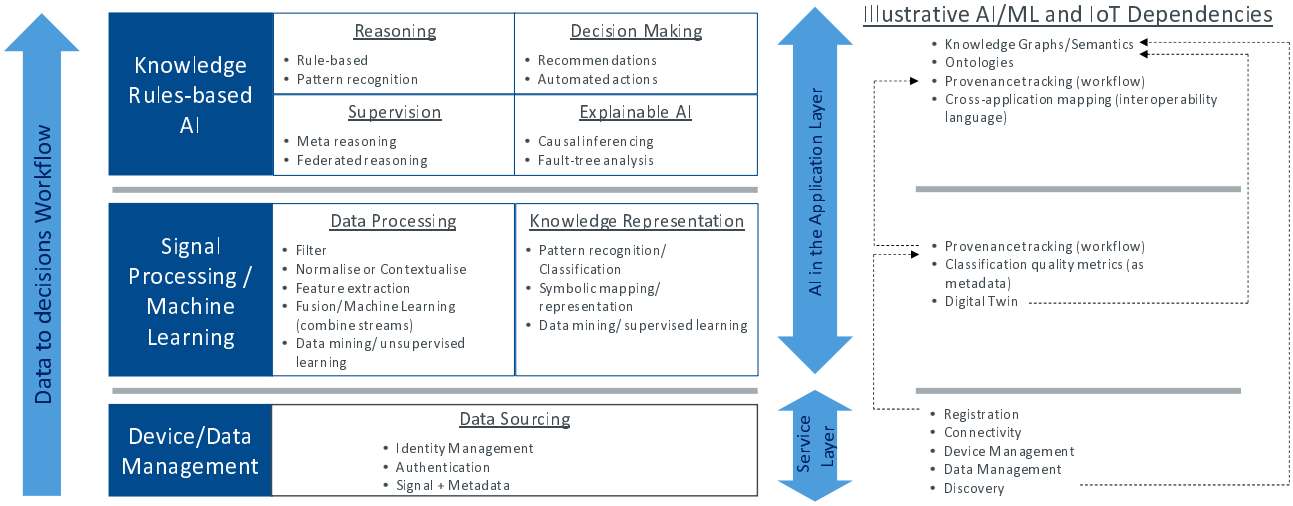

Figure 1: IoT Data to Decisions Workflow

This workflow illustrates some of the generic and commonly used data sourcing and AI/ML capabilities involved in supporting end-to-end IoT solutions. It also shows how AI/ML capabilities depend on service layer capabilities. An example (illustrated in right-hand side of Figure 1) is the relationship between a registration capability that manages the identify of a device and its value in providing information about the provenance of data used in pattern recognition or causal inferencing functions. In this example, the act of tracking data provenance depends on a 'registration' service capability. Provenance tracking can improve the quality and dependability of an AI/ML system and occurs in the background to 'user plane' activity. In architectural terms, such background processes and use of data that improve the quality of AI/ML applications occur in what is referred to as the 'control plane'.

In order to benefit from Artificial Intelligence, the challenge for IoT systems (and the architectures and services that support them) is to make commonly used AI capabilities an integral and standardized part of the IoT solution stack. Developers could then apply such capabilities on user- and control-plane data to improve the performance of individual IoT solutions. One benefit of this approach is to separate the complexity of IoT applications into application-logic and AI-enhancing modules. The availability of generic AI services would avoid the need to re-design and re-develop the interface between applications and devices. It would also improve the prospects for AI-related interoperability between IoT silos and federated systems.

A traditional approach (illustrated in Figure 2) relies on an application ingesting and storing data for processing. The software implementation concentrates AI/ML and service layer capabilities in the application layer. This places a burden on solution developers to master application, AI and service layer disciplines.

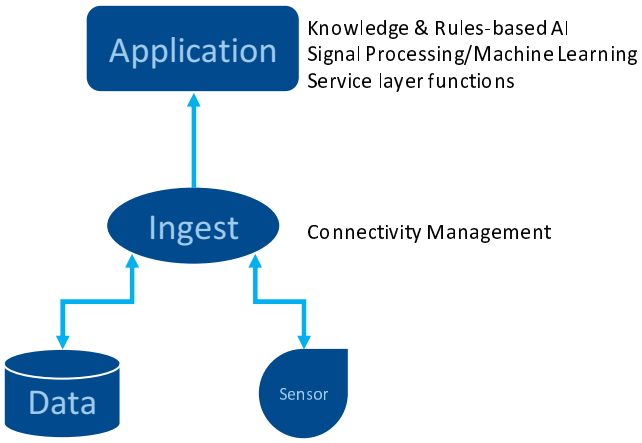

Figure 2: Traditional approach for IoT solutions

The alternative is a developer friendly approach (illustrated in Figure 3). This approach provides developers with an abstraction layer - i.e. a common service layer - that makes AI and the more usual IoT-enabling services accessible through a standardized API. This arrangement means that the IoT application can rely on notifications from the common services layer to trigger its functions when notified of changes in IoT data. It can also draw on a library of AI-enabled services provided within the common services layer.

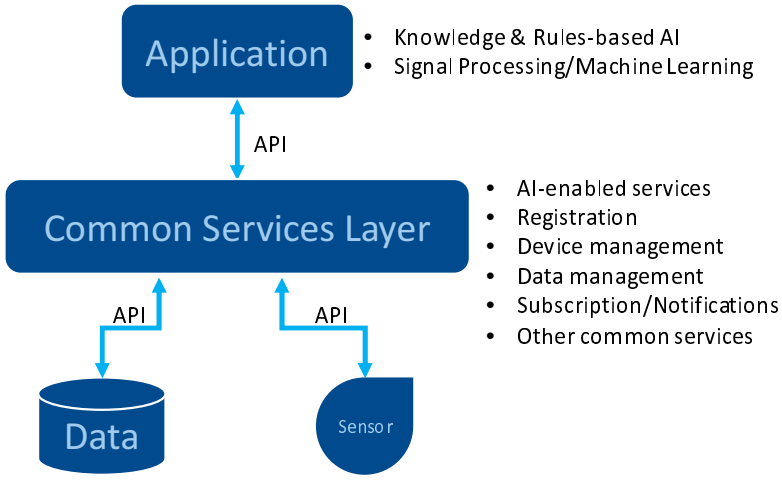

Figure 3: Scalable and developer-friendly approach for IoT solutions

The introduction of AI in typical IoT architectures raises several issues that will be addressed more specifically as part of this study. Some of the issues deal with logical aspects, such as the distribution of action and knowledge, while others consider physical deployment factors.

A three-tier framework to organize the logical aspects of AI in IoT maps AI applications into a 'user' plane. This is where data is collected and used by AI applications to make decisions that trigger alarms or some form of intervention. The services that enable these applications reside in the second layer, which is the 'control' or 'common services' plane. The third layer in this framework is the 'knowledge' plane where supervisory monitoring and whole-system coordination occurs via a separate set of data management and AI functions.

Some of the physical deployment considerations of AI in IoT systems deal with the distribution of capabilities in Fog and Edge architectures as well as in the case of federated systems that need to collaborate across silo boundaries.

4.3 Purpose and content of the present document

The present document addresses the application of Artificial Intelligence to IoT systems, more specifically its impact on existing architectures, in particular on standardized service architectures such as the service layer architecture developed by the oneM2M Partnership Project. Beyond the analysis of the main issues related to the introduction of AI in IoT systems and its illustration with selected use cases, an important objective of the present document is to identify recommendations towards the oneM2M community of potential Work Items that could become part of the oneM2M Work Program.

The target group for the present document is the community of people that is interested in the understanding of the added value of AI in IoT systems and how it can be embodied in standardization work regarding IoT architectures, in particular the Service Layer architecture of oneM2M.

Clause 5 presents the methodology used to analyse the potential impact of AI on oneM2M and addressing the collection and analysis of the most relevant use cases for AI in IoT.

Clause 6 addresses the selection of features chosen from a small set of the use cases considered in this study and which are deemed most informative for the purposes of validation through implementation in the associated Proof-of-Concept (PoC).

Clause 7 summarizes the main impacts of AI on the oneM2M architecture and highlights different approaches that can serve as contributions to the definition of AI-related Work Items for consideration by the oneM2M project.

Clause 8 presents some lessons learned from the above analysis. Based on these lessons, some recommendations are made towards the IoT community regarding standardization as well as some input to be considered for a collaborative effort in oneM2M.

5 Collection and analysis of relevant Use Cases

5.1 Methodology to address the impact of AI on oneM2M

On their own, AI and IoT apply to a broad universe of situation. For this initiative, the focus is on Use Cases (UCs) and capabilities that apply at the intersection of AI and IoT. The approach taken begins with an analysis of a set of illustrative use cases and general capabilities related to AI with broad applicability to multiple use cases. The selection of use cases is not intended to provide an exhaustive analysis. Its purpose is to explore an emerging area of AI-enabling capabilities and to characterize some of the requirements that would enhance the performance of IoT systems.

The methodology (summarized in Figure 4) applied for this initiative is as follows:

- Review of AI and IoT literature in order to develop a list of illustrative use cases.
- Analysis of each use case to identify possible implications on the oneM2M service layer.
- Identify common issues across the different use cases.
- Select a relevant subset of issues to be validated in the Proof-of-Concept (described in ETSI TR 103 675 [i.1]).

Figure 4: Approach to analyse service layer needs and features for PoC evaluation

The analysis of use cases will be done in a way that can support the "Scalable and Developer-friendly" approach advocated in clause 4.2. The main objective of this analysis is to understand different uses of AI/ML in IoT systems and their interdependencies with service layer capabilities. When applied to multiple use cases, a related objective is to identify recurring situations in order to identify candidates for testing and for eventual standardization.

5.2 Collection of Use Cases

The following use cases draw on a variety of sources including research, innovation road mapping and pilot projects.

Another potential source of use cases may be the analysis that has already been undertaken in SDOs. An example of such analysis is done in ISO/IEC SC42 Working Group 4 (Use cases and applications) that is currently developing a standards document (see ISO/IEC CD TR 24030 [i.4]). However, these documents are still under development and not entirely dedicated to IoT use cases.

Use-case analysis represents one pathway to analyse potential service-layer requirements arising from the application of AI to IoT. Since some AI components are applicable to multiple use cases, this analysis also considers general purpose capabilities to identify potential 'cross-cutting' requirements. Table 1 lists a set of use cases to be analysed that can be divided between conventional use cases (1 to 7) and cross-cutting capabilities (8 to 9).

Table 1: List of use cases and capabilities analysed

Number	Title
1	Fault management and isolation for IoT field devices
2	Detection of patterns in video streams
3	Autonomic Management and Control of Communications Networks
4	Supervisory road-network traffic management for journey planning or to manage transportation logistics
5	Occupancy Prediction in Smart Parking
6	Language-based pattern recognition in social media/crowdsourced data
7	Knowledge graphs and semantic reasoning in smart buildings
8	Trustworthy AI
9	Verifiable AI

These use cases and capabilities are analysed in more details in the following clause and address the main aspects of the use case. Some points need to be taken into account:

- The term "*Use Case*" will refer to two kinds of objects: "Use Case" will refer to a set of features at application level; "Capability" will refer more to a cross-cutting view applicable to several UCs (e.g. Trustworthy AI).
- The UC "*description*" focuses on some aspects of the use case. In particular:
  - A Use Case is not meant to be comprehensive and describe all the features that could be associated to it. It is an illustration of a subset of specific aspects relevant to the impact analysis and to the definition of the Proof-of-Concept.
  - The final description used by the PoC may incorporate elements from one or more UC and one or more capabilities.
- The "*key characteristics*" refer to the main AI-related features/technologies / that AI could bring to the development of the UC.
- Some "*potential benefits*" (e.g. opportunities for extension/improvement) can be identified and further analysed.

5.3 Initial analysis of the Use Cases

5.3.1 Fault management and isolation for IoT field devices

5.3.1.1 Description

Fault detection aims to identify defective states and conditions within computing systems, subsystems and components and ensure their proper functionality to reduce their rate of deterioration, hence better customer experience. There is a need to be an effective maintenance service in place to ensure that IoT devices are running at their best. The inputs of maintenance services are measurements reflecting the health state of the monitored item.

In this use case, an IoT module will be prototyped for fault detection and isolation of IoT device data in a smart building environment using both a rule-based fault detection and a self-learning fault detection algorithm based on e.g. statistics sliding window approach. The rule-based approach would be based on available manufacturer datasheets including rules if available. The self-learning algorithm is based on determining trend vectors and comparing such vectors with longer term historic data.

Figure 5 provides a summary of the associated architecture.

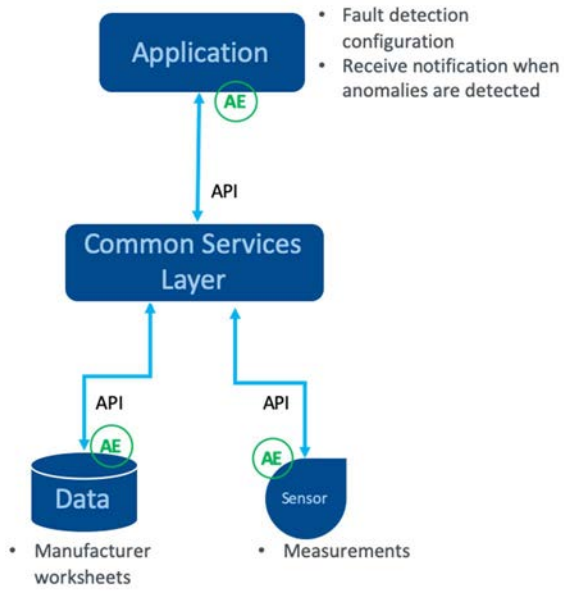

Figure 5: Use case 1 - Architecture

5.3.1.2 Key observations

- Rule-based (data loss, spikes, crossing of thresholds, etc.)
- Supervised learning
- Data-centric detection (outliers, spikes, …)
- Linear and non-linear algorithms
- Exponential smoothing
- Auto-regressive Integrated Moving Average (ARIMA)

5.3.1.3 Implications for oneM2M

Potential benefits for oneM2M:

- Introduce new fault detection functions (Rule-based/ML-based) on the Common Service Entity ready to be configured and used by an Application Entity, subject to access control policies, to detect anomalies about collected sensor measurements and receive the corresponding notifications.

oneM2M dependencies:

- Data management: Extend oneM2M resources (e.g. containers, flex Containers, etc.) with fault detection attributes.
- Application and Service Layer Management: Extend the oneM2M MCA interface to support configuration of fault detection parameters (CRUD).
- Discovery: Discover resources based on fault detection attributes (Extend filter criteria with fault detection attributes).
- Subscription and Notification: Notify an application when an anomaly is detected.
- Security: Define dedicated access control policies related to fault detection, creation, configuration and notification.

New AL/ML CSF requirements:

- Rule-based fault detection CSF: configurable service that detect faults based on predefined rules and notifies an application when it detects anomalies in the data (data loss, spikes, crossing of thresholds, etc.)
- ML-based fault detection CSF: configuration service that detect faults based on ML algorithms (e.g. exponential smoothing, auto-regressive Integrated Moving Average (ARIMA), etc.) and notify an application when it detects anomalies in the data.

5.3.2 Detection of patterns in video streams

5.3.2.1 Description

Detection of patterns in video and camera streams enables users to identify scenes, objects, and situations in images uploaded to the service using aims visual recognition based on Artificial Intelligence and Machine learning. Subjects and objects contained in an image are automatically identified, organized and classified into logical categories in order to provide add high added value services in cities such as car vandalism and fire detection.

In this use case, an IoT module will be prototyped for images classification using machine learning and trained data. The IoT module supports multiple classifiers: predefined and custom models. A camera agent will be developed to quickly test the proposed prototype and simplify the integration with real devices within the city. The camera agent reads periodically images from the disk and push them to oneM2M platform. The images could be provided by a real camera or any other external sources.

Figure 6 provides a summary of the associated architecture.

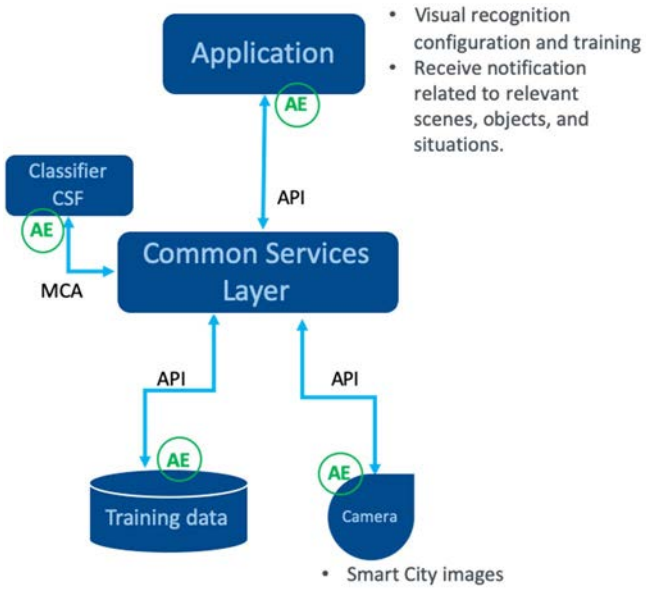

Figure 6: Use case 2 - Architecture

5.3.2.2 Key observations

- Image classification (predefined classifier / custom classifier)
- Object detection
- Object tracking
- Semantic Segmentation
- Instance Segmentation

5.3.2.3 Implications for oneM2M

Potential benefits for oneM2M

Introduce visual recognition functions (predefined classifier/custom classifier) on the Common Service Entity ready to be, configured trained and used by an Application Entity, subject to access control policies, to identify relevant scenes, objects, and situations within the city and receive the corresponding notifications.

oneM2M dependencies:

- Data management: use existing oneM2M resources (e.g. containers, flex Containers, etc.) or create new ones to store training images, real images coming from cameras, and the result of the classification.
- Application and Service Layer Management: Extend the oneM2M MCA interface to support the configuration and train of the visual recognition service (CRUD).
- Discovery: Discover resources based on the visual recognition attributes (Extend filter criteria with image classification attributes, types, etc.).
- Subscription and Notification: Notify an application when a car vandalism is detected within the city.
- Security: Define dedicated access control policies related to visual recognition creation configuration and notification.

New AL/ML CSF requirements:

- Predefined-classifier CSF. The CSF comes with a predefined and pretrained classifier for Object detection, Object tracking, Semantic Segmentation, Instance Segmentation, etc.).

- Custom classifier CSF image classification CSF. Allow an application to create its own classifier and train it to implement specific visual recognition use cases.

5.3.3 Autonomic Management and Control of Comm. Networks

5.3.3.1 Description

This use case deals with the application of AI/ML to enable self-configuring, self-healing, self-optimizing, self-protecting capabilities in communications networks. The concept can also be extended to federated networks where there might arise a need for individual networks to cooperate in the transfer of IoT data.

Figure 7 illustrates an example of a simple network configuration that might be deployed in a smart home or smart building environment. The application involves data supplied by groups of IoT devices that communicate via Wi-Fi and Bluetooth networks. Wi-Fi connected devices rely on a home gateway whereas Bluetooth devices, which are treated as being movable, can transmit data either via the home gateway or a smartphone. The application that uses IoT data occasionally needs to integrate data from the Bluetooth devices that may arrive via one of two paths. Over a given time interval, this may require it to assemble a data sequence by combining segments of data that arrive via two separate communications channels. This becomes possible if the home gateway and smartphone use common service layer capabilities to interoperate.

This use-case is an example of a wider class of use cases that allow communications networks to operate in parallel and where there are benefits in coordinating behaviours to manage the use of resources and to enable fault-tolerant operations. Parallel networks can exist as separate physical instances, as illustrated above, or as virtual instances as in the case of network slices.

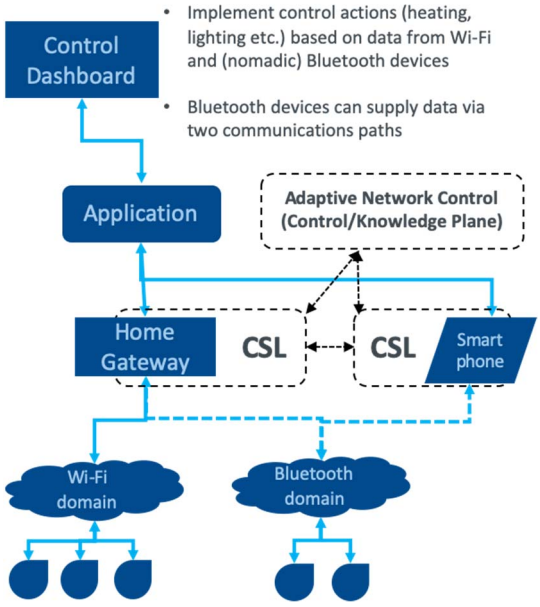

Figure 7: Use case 3 - Architecture

5.3.3.2 Key observations

The application of AI/ML In this example scenario relies on interactions between the control- and knowledge planes. The common services that manage the flow of (user-plane) data manage the identity of IoT devices and data associated with them. Within the knowledge plane, other common service functions create a digital twin representation of the data supply chain by drawing on discovery and semantic modelling capabilities. Once data beings to flow, this digital twin provides the framework for delivering a high-quality data stream. It might trigger actions to prioritize data between Wi-Fi and Bluetooth channels based on signal quality. Other capabilities might be used to reconstitute data that arrives by different channels over time and to supplement the data stream with qualitative meta-data such as the media used in its collection over time.

5.3.3.3 Implications for oneM2M

Potential benefits for oneM2M:

- Combine resource discovery and ontology capabilities to represent dynamic communications networks (digital twin), applying information about the autonomic capabilities of individual components to improve the quality of data handling.

oneM2M dependencies:

- Registration - required to use CSL services.
- Data management - data storage with potential enhancements to reconstitute data from intermittent transfers or transfers occurring over parallel channels.
- Discovery - allows applications to find CSL resources.
- Subscription/Notification - define events of interest allowing CSL to generate notifications.
- Group management - supervision of Bluetooth and Wi-Fi domains.

New AL/ML CSF requirements:

- Rule-based handling of data flow depending on availability of home gateway or smartphone devices to capture Bluetooth domain data. This could involve:
  - prioritisation of data depending on connectivity quality of different channels.
  - reconstitution of data that may be collected from different channels over a period of time.
- Model-driven reasoning for explain-ability.

5.3.4 Supervisory road-network traffic management for journey planning or to manage transportation logistics

5.3.4.1 Description

This is an example of a multi-layer use case. It is set in the context of a journey planning system for a regional, road transport network and where IoT capabilities are used to monitor traffic flows at three road junctions (this is illustrated in Figure 8). Localized applications maintain a record of traffic flows and predicts an estimate of congestion based on current data and historical patterns; Application #1 handles Zones a and B while Application #2 deals with a more distant Zone C. The role of Application #3 is to support regional journey planning.

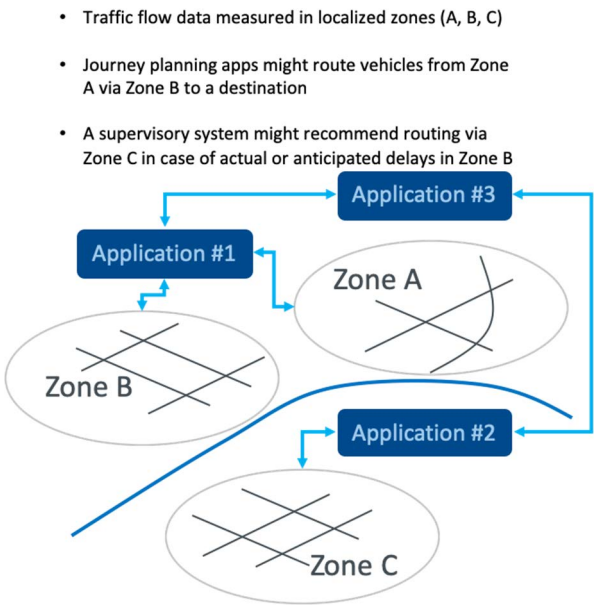

Figure 8: Use case 4 - Architecture

An example scenario might involve a journey that either passes through Zones A and B or the alternative of a longer trip that involves traveling via Zone C. In this case, Application #3 deals with a higher layer of reasoning compared to Applications #1 and #2 although it relies on intelligence supplied by these two applications. An example scenario might involve a localized disruption that causes journey planning applications to reroute traffic via a less congested route. This would trigger a supervisory override which would anticipate a second-order congestion effects if the new route is known to lack the capacity to handle diverted vehicles. The journey planning application could take this new information into account before recommending a route or advising the use to express a preference based on delay probabilities.

Potential scenarios involve transportation in smart cities or smart regions as well as applications in the handling of parts for manufacturing assembly lines.

5.3.4.2 Key observations

This use case illustrates a class of cooperative and multi-level AI/ML applications that involve different user viewpoints (e.g. local traffic situation and regional picture) and the value of a common representation for AI/ML reasoning purposes. This common representation depends on basic CSFs (e.g. identity management and discovery) and novel applications of discovery and semantic CSFs to create a logical framework for cooperative AI.

5.3.4.3 Implications for oneM2M

Potential benefits for oneM2M:

- Enable federated reasoning at the level of user applications by providing a system-wide view of complex networks.

oneM2M dependencies:

- Federated IoT Platforms.
- Registration - required to use CSL services.
- Access Control Policies to grant/deny service requests.
- Data management - data storage.
- Discovery - allows applications to find CSL resources.
- Subscription/Notification - define events of interest allowing CSL to generate notifications.

New AL/ML CSF requirements:

- Discovery and Semantics - to enable cross-application collaboration.
- Look-ahead prediction.

5.3.5 Occupancy Prediction in Smart Parking

5.3.5.1 Description

A common use case for Smart Cities is Smart Parking, where, through different means, such as sensors, gates or cameras, on-street or off-street parking areas can be monitored in real-time. This helps city officials in understanding the current status of the parking in the city. By storing this data, it is possible to also analyse it and understand where more or less parking is needed, as well as how different conditions (e.g. weather, events) impact the parking occupancy. Through this data, it is also possible to apply ML methods to predict how parking occupancy will evolve in the future, and how it will be affected by certain events.

In this use case, a prototype of an IoT module will be developed for parking occupancy prediction through ML methods, the predictions will be based on data that could be obtained from sensors, APIs or camera feeds.

Figure 9 provides a summary of the associated architecture.

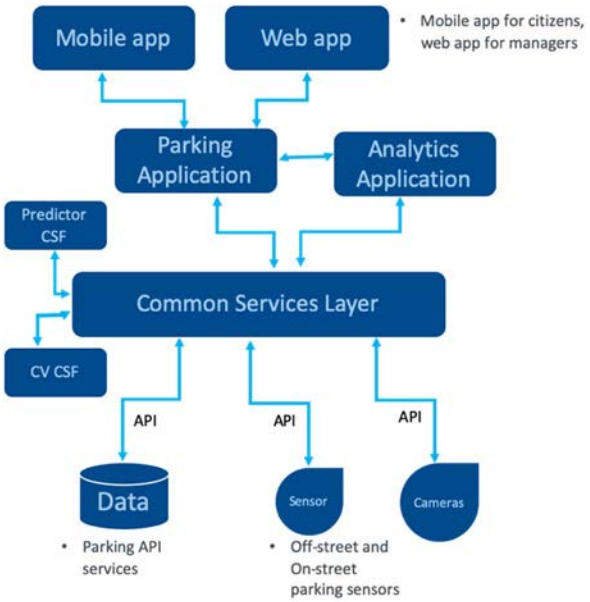

Figure 9: Use case 5 - Architecture

5.3.5.2 Key observations

- Data harmonisation
- Semantic modelling
- Data access/authorization
- Multi-layer AI
- Supervised learning
- Object detection

5.3.5.3 Implications for oneM2M

Potential benefits for oneM2M

Introduce prediction methods on the Common Service Entity ready to be configured, trained and used by an Application Entity to make predictions on the occupancy of parking areas. Introduce visual recognition models, either already trained or the ability to train them, on the Common Service Entity ready to be configured, trained and used by an Application Entity to identify objects (in this particular case, cars) in parking areas. Introduce alert generation methods that can be configured and used by an Application Entity to generate alerts based on incoming data and predefined rules.

oneM2M dependencies:

- Security: to manage access to own parking sensors
- Discovery: to discover other parking information that might exist
- Semantics: to understand the information
- Data Management: to manage the data obtained from different sources (harmonise it)
- Device Management: to manage the various parking sensors (mostly to know if they are working or not)
- Semantics: to perform semantic mapping in order for parking information from different systems can be understandable

New AL/ML CSF requirements:

- Feature extraction to detect problems in sensors (unusual values, lack of communication, etc.)
- Alert generation based on predefined rules
- Common ML methods (ARIMA could be straightforward, others not so much), pre-trained CV models (e.g. for car detection, etc.) or ability to train own ML/CV model

5.3.6 Language-based pattern recognition in social media/crowdsourced data for occurrences classification

5.3.6.1 Description

In Smart Cities, data can come in many forms and from different sources. The common presented use case tends to be demonstrated through sensors, but data obtained directly from citizens, through what they publish on social-networks or provided through other means (e.g. mobile apps to report occurrences) can be very insightful and valuable. However, this data is not structured, and its quality can be questionable (e.g. due to typos, ambiguity or incompleteness). As such, special care has to be taken in order to obtain insights from it. When done properly, it becomes a useful source to know what is happening throughout the city in a relatively inexpensive way.

In this use case, a prototype of a module has been developed to process, through Natural Language Processing methods, incoming text data from users, obtained from social networks or mobile apps, in order to detect occurrences and location of disasters throughout the city. Common NLP modules have been used to process the text and take valuable data out of them.

Figure 10 provides a summary of the associated architecture.

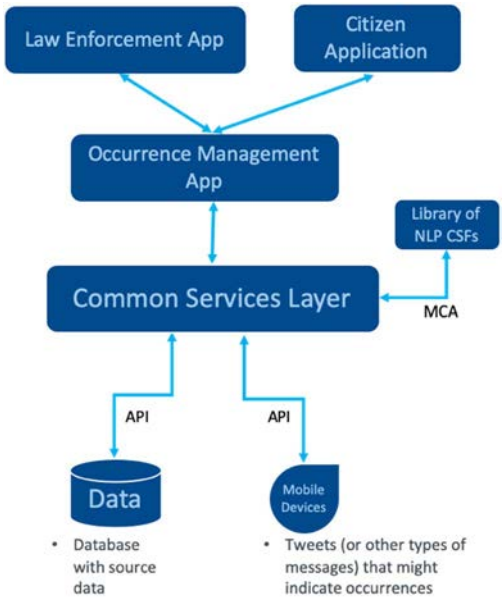

Figure 10: Use case 6 - Architecture

5.3.6.2 Key observations

- Data access/authorization
- Workflow management
- Multi-layer AI

5.3.6.3 Implications for oneM2M

Potential benefits for oneM2M:

- Introduce common NLP methods (e.g. tokenization, name-entity recognition) on the Common Service Entity ready to be used by an Application Entity.

oneM2M dependencies:

- Discovery to detect sources of data
- Data management to process the extracted data
- Security to restrict access to private data and make sure the data is not tampered with

New AL/ML CSF requirements:

- Feature extraction through common NLP tasks such as Tokenisation, PoS Tagging, Name-entity recognition, Topic modelling, Speech recognition

5.3.7 Knowledge graphs and semantic reasoning in smart buildings

5.3.7.1 Description

This use case is an example of situations where AI/ML capabilities are applied to provide second-order insights and decision support solutions. Consider the example of a smart building that contains one application to handle energy production (e.g. roof mounted solar and wind powered generating capacity) and another to handle energy consumption by providing building tenants with recommendations to optimize efficiency. Each of these applications can apply AI/ML capabilities to user-plane IoT data. There is scope to combine information about the interaction of these two applications in support of a third application that performs fault diagnosis to support the building manager when problem situations arise. This arrangement is illustrated in Figure 11.

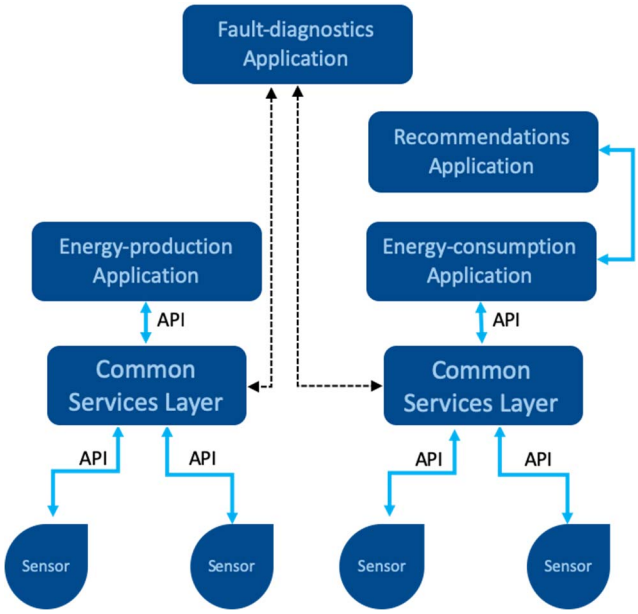

Figure 11: Use case 7 - Architecture

5.3.7.2 Key observations

The application of AI/ML capabilities in this example relies on the formulation and use of knowledge graphs that represent entity relationships between building systems and tenant properties to model cause-effect dynamics. The primary use of AI/ML would be in providing recommendations, going one step beyond visualization dashboards, to tenants in order to improve overall energy efficiency.

The secondary application of AI/ML would take the form of knowledge graphs for fault detection, using reasoning approaches. In addition to the use of semantic models, this capability would rely on the ability to combine ontologies from different domains through a common language mapping process in order to create digital-twin equivalents for the purpose of reasoning and explainable AI.

5.3.7.3 Implications for oneM2M

Potential benefits for oneM2M:

- Enable cross-silo sharing of IoT data and application resources to create a system-wide view of applications and their interdependencies to optimise system behaviour or to detect and diagnose faulty operations.

oneM2M dependencies:

- Registration - required to use CSL services
- Access Control Policies to grant/deny service requests
- Data management - data storage

- Discovery - allows applications to find CSL resources
- Ontologies for different domains
- Common language mapping for cross-domain ontologies

New AL/ML CSF requirements:

- Model-driven reasoning for explain-ability
- Causal reasoning (back tracing/fault tree analysis) and data integrity testing - hierarchical model of data sources based on device profile data

5.3.8 Trustworthy AI

5.3.8.1 Description

The purpose of this use case is to address a general-purpose capability in IoT solutions related to trustworthiness. This is analogous to the provision of security services that can be added to any IoT application and tailored to the circumstances of each use case.

The field of trustworthiness in ICT systems is broad and its definition is beyond the scope of this project. For the purposes of exploring its AI/ML-related impact on IoT solutions, this analysis focuses on two aspects of trust. One of these concerns profile data for any given resource. In the case of a connected device, this might include details about the supplier, information about its engineering quality (e.g. certification conformance, adherence to industry design guidelines, etc.) and information about its security capabilities. This is not a comprehensive list of attributes, but it serves to illustrate the value of profile data.

The second class of data related to trust concerns time-history information. In the case of devices/sensors this could include a report on up-time statistics and regularity of firmware and security credential updates. In the case of AI/ML applications, trustworthiness data could be based on dependability and accuracy statistics. Examples might involve tracking predictive or pattern recognition/classification accuracy over time.

Figure 12 illustrates how trustworthiness might be provided as a common service function in a simple IoT solution. The CSF would construct a record of trustworthiness data (profile and time-history performance) that the application or a user could query as needed. It could maintain a record for the sensor, the application and another AI/ML CSF such as a 'classifier' as illustrated.

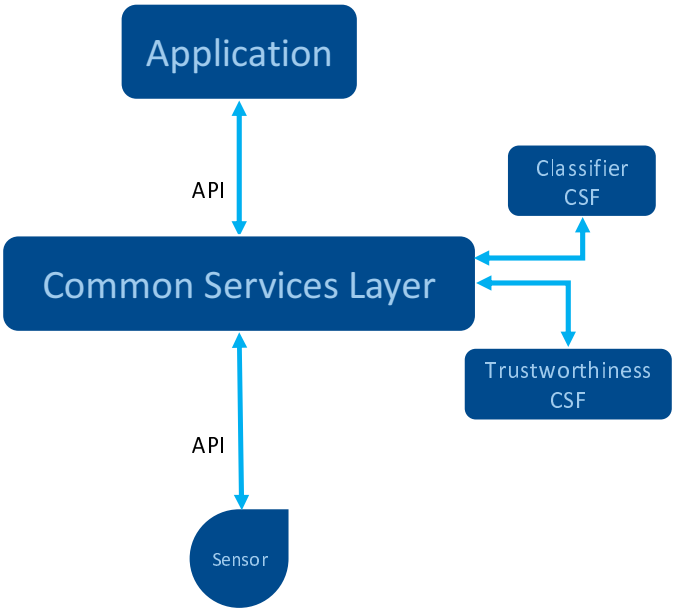

Figure 12: Use case 8 - Architecture

5.3.8.2 Key observations

The application of AI/ML capabilities in this example relies on managing the identity and trust attributes for components in the IoT solution stack (user plane) as well as the performance of AI/ML capabilities provided as CSFs. Responses to queries about trustworthiness could take the form of quality indicators (e.g. low-medium-high integrity or, improving/worsening predictive accuracy using fuzzy logic principles) or confidence estimates (e.g. on a 0 % to 100 % scale).

5.3.8.3 Implications for oneM2M

Potential benefits for oneM2M:

- Use service layer (control plane) data to report on the trustworthiness of elements in an IoT solution and to assign a degree of trustworthiness for AI/ML predictions or classification results.

oneM2M dependencies:

- Registration - required to use CSL services
- Data management - data storage adapted to record data related to trustworthiness
- Discovery - allows applications to find CSL resources
- Subscription/Notification - define events of interest allowing CSL to generate notifications

New AL/ML CSF requirements:

- Resource data - profile information for devices/sensors (manufacturer, certification compliance information, security capabilities, etc.)
- Time-series record for data sources covering parameters related to trust e.g. up-time performance, self-reported malfunctions, firmware status (are devices patched promptly?)

5.3.9 Verifiable AI

5.3.9.1 Description

This use case is another example of a general capability that has applicability to a variety of AI/ML IoT use cases. It deals with an emerging field of study concerned with the provision of a measure of quality (or correctness) as to whether an AI algorithm is performing correctly. It can involve the application of formal verification techniques from the software domain.

For the purposes of this analysis, two methods for verifying an AI/ML application are envisaged. One of these provides a means of tracing an allocation layer decision back to the data inputs and logic that produced it. At its most basic, this corresponds to a form of fault tree analysis.

A second approach to verify an AI/ML result is to corroborate it by reference to other sources of data. Consider the example of a sensor that detects when it is raining. An application might verify its status by checking whether images from a camera show that it is raining or that pedestrians are either carrying unfurled umbrellas or wearing raincoats.

Figure 13 provides an example of a possible use case. The common components are the three layers of an IoT solution stack. This arrangement is augmented by two CSFs. One of these applies semantic modelling techniques to create a digital twin on the IoT solution. The second functions as a fault-tree analyser.

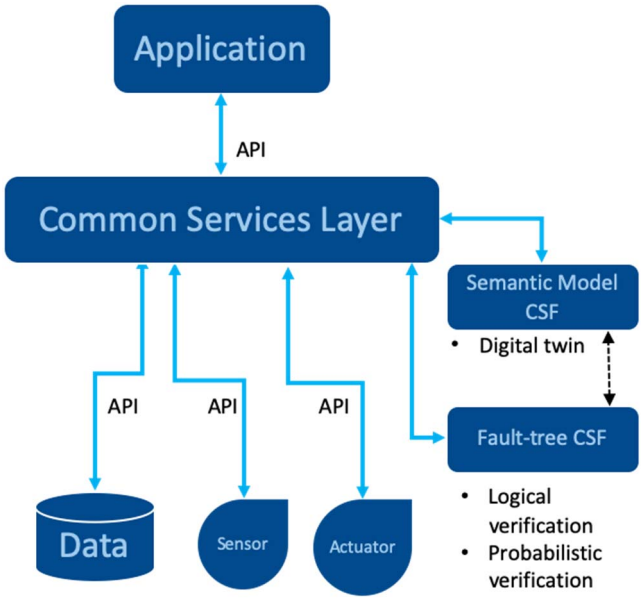

Figure 13: Use case 9 - Architecture

5.3.9.2 Key observations

This example of verifiable AI relies on the creation of a digital twin of the IoT application using resource discovery and semantic modelling capabilities. The digital twin may be configured at the design stage. In a technically advanced situation, it may be possible for a CSF to create a digital twin by learning about the components of the IoT solution. Once there is a need to verify the correctness of an application output, a user or the application could apply fault-tree analysis techniques to trace the elements that contributed to the decision. This might take account of issues such as data recency if some devices report their status periodically (e.g. every hour) while others feed data on a continuous basis.

Another use case example would involve verifying the readings provided by a sensor by checking the consistency of its values against readings from other sensors. This would depend on configuring an AI/ML verification algorithm using prior knowledge. In the case of a combustion engine, for example, a verification process would involve cross checking whether the rate of fuel supply, engine speed and exhaust temperatures are within consistent operating envelopes.

5.3.9.3 Implications for oneM2M

Potential benefits for oneM2M:

- Potential extensions of the semantics CSF to support causal reasoning by creating digital twins (knowledge graphs) of end-to-end solutions.

oneM2M dependencies:

- Registration - required to use CSL services
- Data management - data storage adapted to record data related to trustworthiness
- Discovery - allows applications to find CSL resources
- Semantics

New AL/ML CSF requirements:

- Trace-back (control-plane) of resources that contributed to a data-driven decision (user-plane)
  - Logic chain
  - Probabilistic measure of correctness

- Discovery of other data or applications that can corroborate the outputs from a solution
  - E.g. verification based on benchmark or historical comparisons with other sources
  - Reasonableness test by comparing outputs using two methods of calculation

6 Main components for the Proof-of-Concept

6.1 The Proof-of-Concept

The Proof-of-Concept (PoC) is meant to validate some of the approaches that have been underlined in the analysis of Use Cases done in clause 5. Considering that there are different objectives regarding the architectures chosen for the provision of an AI-based services as well as different implementation platforms available, the PoC activity covers the implementation of three different Use Cases. These are The Use Cases UC1, UC2 and UC6 as addressed in clause 5.

The three PoC Use Cases and their objectives are described below.

6.2 The PoC Use Cases

6.2.1 PoC Use Case 1: Fault management and isolation for IoT field devices

There are many reasons why an IoT device might not render the services it was designed for. These include reasons related to design errors, implementation errors, human operator errors, wear, aging, and environmental aggressions. A faulty IoT system could quickly run out of control and become the source of energy and material waste, loss of production, environment damage, and in some worst-case scenarios the loss of human lives.

A fault is defined as an unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable condition. A fault might occur suddenly, with catastrophic results, or it could occur gradually over time with different patterns of behaviour including a slow degradation that might exhibit one of the following patterns: ramp, exponential, parabola, or pulses.

For the purposes of this PoC use case, an IoT module undertakes fault detection and isolation of IoT device data in a smart building environment. IT uses both a rule-based and a model-based fault detection algorithm based on a statistical, sliding window approach. This focus of this use case is specifically about measurements, rather than device properties, and how they can be dealt with and implemented in oneM2M.

The rule-based approach is based on available manufacturer datasheets including failure-reporting rules if available. The model-based approach relies on a ML algorithm determining trend vectors and comparing such vectors with longer term historic data. The system model may be mathematical, or knowledge-rules based.

The main goal is to extend oneM2M with fault detection capabilities to make it possible for oneM2M developers and users to detect malfunctions in real time, as soon and as reliably as possible. The next step is to find the root cause by isolating the system component (IoT devices, application logic, etc.) whose operating mode has deviated from normal. The final step is to estimate the size and type or nature of the fault (which can be interpreted as a form of explainability). The automatic isolation of faulty devices and applications natively in oneM2M can be done through a notification message (the simplest approach), or through a real isolation from the platform (which is more complex).

The PoC provides oneM2M with the architecture and software mechanisms which allow it to achieve a given objective not only in normal operation, but also in given fault situations.

6.2.2 PoC Use Case 2: Detection of patterns in video streams

Visual recognition represents a relative understanding of visual environments and their context involving many academic subjects, such as computer science, mathematics, engineering, physics, biology, and cognitive science.

In this use case, an IoT module undertakes image classification using machine learning and trained data. The IoT module supports multiple classifiers including predefined and custom models. This focus of this use case is specifically about images and how they can be dealt with and implemented in oneM2M.

A camera agent is developed to quickly test the proposed prototype and simplify the integration with real devices within the city. For testing purposes, the camera agent reads images periodically from a data repository and pushes them to the oneM2M platform. In an actual deployment scenario, images would be provided by a real camera or some other external source.

The main goal here is to extend the oneM2M architecture with visual recognition capabilities to make it possible for oneM2M developers and users to gain high-level understanding from digital images or video streams through the construction of explicit, meaningful descriptions of physical objects, and scenes from images or video streams and use them to make relevant decisions.

6.2.3 PoC Use Case 3: Language-based pattern recognition in social media/crowdsourced data for occurrences classification

Feedback provided by humans is a useful source of data for multiple applications (such as Open311 https://www.open311.org/ for civic issue tracking). This feedback usually comes as text and can be originated in different sources such as mobile applications, social media networks or voice messages shared with mobile devices or smart speakers, just to name a few. Since the content is not generated by a programmed entity (such as a sensor or a CCTV camera), in order to be used by an AI agent, it has first to be checked. This is to ensure that content is not incomplete, does not contain typographic errors or is somehow ambiguous or incoherent.

In this use case, an IoT application shares text messages with an oneM2M CSE, which then performs the task of cleaning up the received text (i.e. corrects typographical errors) before sharing it with an application that classifies the text (if it is an occurrence or not). Both steps in this process make use of Natural Language Processing tools.

The implementation was done and validated with a combination of OpenMTC (an open source oneM2M platform) and the open-source broker from FIWARE. It required two AEs (AEs) and one Common Service Entity (CSE) with new Common Service Functions (CSFs) to prepare text for proper analysis. While one of the AEs simulates a device sharing text generated by humans, based on content provided in social media networks, the other leverages Natural Language Processing tools to classify the text received as civic occurrences.

The main goal was to enrich (and extend) CSF capabilities within a CSE (gateway) through the use of NLP and data clean-up services, going beyond the connectivity and service instantiating capabilities that are core to the oneM2M standard. Though this arrangement, data processing occurs closer to the devices or IoT applications producing or sharing textual data.

7 AI benefits and impact on the oneM2M service layer

7.1 Introduction

A first area for future work is the enhancement of existing CSFs so that they provide additional data to improve the performance of AI/ML algorithms. This could involve filtering to detect and report anomalies in sensor data. With growing interest in trustworthiness, verification, and security a second area for improvement involves the life cycle aspects of sensors and data. For example, the profile of a sensor and updates to its firmware and security credentials over time reflect its trustworthiness. Similarly, the ability to trace the provenance of data in a complex application that may involve data sourced from several organizations has a bearing on the quality of decisions derived from that data. Existing CSFs can be augmented to provide this level of tracking.

The analysis of AI/ML use cases in clause 5 identified some recurring patterns that justify standardization to provide developers with re-usable capabilities at the common service layer. Consequently, another area for future work is to add new modules to the family of oneM2M's common service functions. These modules would provide AI/ML capabilities as callable services. An example might be a pattern recognition CSF that could be configured for a specific class of problems (e.g. detect pedestrians, detect if a car has been vandalised, etc.).

7.2 Enhanced existing CSFs

Existing CSFs could be enhanced to support new capabilities based on AI and ML in the oneM2M architecture. This could be done, for example, by defining new attributes and metadata in existing resources to pave the way for the AI-Based processing.

The PoC developed for UC1 (Fault management and isolation for IoT field devices) extends the Container resource attributes with new metadata related to IoT device monitoring. This serves to identify when a fault has occurred and pinpoints the type of fault and its location. Two categories of fault detection techniques could be then considered:

- Rule-based: A fault is said to be detected if a measured discrepancy exceeds a certain threshold.
- Model-based: A comparison of the discrepancy between a set of sensor readings and a set of expected values, derived from a model or 'digital twin', could highlight different patterns of behaviour, each pattern corresponding to a normal or one of several different faulty conditions.

The oneM2M architecture already supports a large list of CSFs. The addition of new capabilities does not necessarily require the definition from scratch of a new CSF where small enhancement across existing CSFs may be feasible.

7.3 New CSF callable as a service CSF

UC2 and UC3 use a remote CSF as a callable service and for which the following observations are applicable:

- UC3 (Language-based pattern recognition in social media/crowdsourced data for occurrences classification). oneM2M's architectural framework supports distributed IoT systems, with the potential to host CSFs both in the Cloud and in Edge devices. One of the benefits of AI in relation to the oneM2M layer is to enable the analysis typically performed by AI agents to be run closer to the devices and sources of the information. Furthermore, it allows for quick and scalable replicability of the common service functions for different AI use cases that involve Natural Language Processing (NLP). Examples of this include the autonomous processing of input voice messages linked to smart home devices or running chatbots as oneM2M applications for different application purposes.
- UC2 (Detection of patterns in video streams). This use case has been developed with the intention of illustrating a new CSF as a callable service. It has shown that, in some cases, the addition of a new CSF may also require the enhancement of existing CSFs to run the new service and to maintain the consistency of the oneM2M architecture. For example: extending the communication binding, enhancing the security policies, or enriching the MCA interface to exposes new type of resources and attributes, or more adapted serialization formats.

The creation of a new Common Service Function (CSF) as a callable service to enrich an oneM2M architecture with Artificial Intelligence capabilities is expected to bring the following benefits for the users of oneM2M:

- The possibility to integrate existing AI/ML packages available on the market to create mixed-technology IoT systems.
- Through oneM2M's abstraction properties, it is possible to provide developers with greater flexibility in using different technologies in the IoT stack as well as increased resiliency, and scalability.
- Developers can create new services organized around business logic and capabilities.
- Through careful architecture design, it becomes possible to localize complexity and to compartmentalize knowledge.
- The use of oneM2M resource labelling would improve the discoverability of AI Agents.
- Parallelization of tasks (agents) to which several oneM2M applications can subscribe.

7.4 Expanding the oneM2M capabilities

7.4.1 Direct use of oneM2M interfaces and resources

In the current version of oneM2M standard, developers wishing to use specific features (e.g. for fault detection management and isolation) have to develop their own module. This may be costly in terms of the learning process, time, and money, resulting in several fragmented, not-interoperable pieces of software. Extending oneM2M capabilities to natively support use cases such as UC1 or UC2 makes it possible for developers to use oneM2M directly, via the MCA interface, as well and existing resources (e.g. to configure fault management policies as in UC1) in a standard way and share them among other applications. This arrangement would allow developers to remain focused on the business logic of their system.

7.4.2 Generic methods

The main outcome of UC3 is to demonstrate how the oneM2M CSE can provide generic methods that can be applicable in many AI applications. The particular case of UC3, which focuses on textual data, has many examples of usage that are generalizable from this use case. Text cleaning can be useful in small sets of textual data (e.g. SMS, chatbot interactions, tweets) or large pieces of textual data (e.g. word documents, outputs of optical character recognition services), and its usefulness goes beyond AI. For this particular topic, such a CSF can always be useful, since textual data can be ambiguous, incomplete, or incorrect, which makes it more difficult to collect insights using common AI methods. Another potential impact of this PoC is the usage of the oneM2M as part of Big Data architectures as a core component of ETL tasks (Extract-Transform-Load) and clean-up of information before its processing and analysis.

8 Conclusions

8.1 Lessons learned

The analysis of the use cases in clause 5 shows that the range of potential applications of AI in IoT is very large and that virtually all applications can be AI-enhanced. However, faced with multiple enhancement scenarios, applications developers can rapidly be overwhelmed by the burden of maintaining a large variety of AI-based modules, data models and data sets. This is one rationale for relying on an intermediate service layer that provides access to a larger and simple to use library of AI-enhanced components.

The development of typical use cases for the purposes of the Proof-of-Concept phase of work shows that it is relatively straightforward to expand the oneM2M architecture and component sets with AI capabilities. Various possibilities have been explored and demonstrated as feasible (i.e. enhancement of existing CSF, new CSF callable as a service, and a mixture of both).

The oneM2M framework which has been designed and implemented for IoT scenarios, fits very well when creating AI use cases over it. Its distributed nature makes it easy to distribute incoming workloads over several agents, thereby raising their scalability and performance.

These considerations showcase the possibilities and the benefits that would be offered to application developers using enhanced oneM2M platforms. The following clause provides input for future discussion and consideration in oneM2M.

8.2 Input for discussion in oneM2M

8.2.1 Standardization approach

Current oneM2M standardization activities are dealing with topics such as data interoperability, data licensing and the expansion of verticals addressed by the SAREF ontology (ETSI TS 103 264 [i.6] and https://saref.etsi.org). AI and ML are complementary to these other technologies but may not be being addressed. On-going work in the oneM2M TP should explicitly address the AI/ML implications through Requirements & Domain Modelling activities. This may involve new work items or the addition of an AI/ML-implications assessment as part of existing work items.

A second observation about AI is that the current industry focus is on applications of AI in the 'user' plane whereas the longer-term issues relate to enablers of AI, which play a role in the 'control' and 'knowledge' planes. These enablers can improve the quality of AI sub-systems in IoT systems. They can also deliver new capabilities about the quality of AI applications by enabling the flow of IoT data and behaviour of data models to facilitate explainable, trustworthy, and verifiable AI. The general approach in this case is to publicize use-case examples and to initiate a discussion about these emerging requirements, within oneM2M and the wider IoT community. Experienced members of oneM2M can also help to identify which CSFs or resource-model elements within the existing technical specifications make such solutions possible.

8.2.2 Expected properties for oneM2M-based services

8.2.2.1 Horizontality

The modular nature of the oneM2M architecture makes it highly extensible by extending existing CSFs and creating new ones to support new and general-purpose capabilities at the abstraction layer. AI an ML based services may solve very specific problems, rely on specific training data, and require fine tuning to work. As a result, the whole service can become tightly coupled to a particular domain or use case. oneM2M is a horizontal architecture and remains so. It is therefore important to adapt any domain-specific service function in a generic way before implementing it as a CSF within oneM2M architecture.

8.2.2.2 Simplicity

Compared to procedural (classical) services defined as a simple succession of steps subject to simple conditions, AI and ML based services seek not to solve a simple process but to solve what are called complex system problems. For this reason, AI algorithms could be become themselves dangerously complex and ML techniques may fail to reveal the underlying logic and physical connotations of the problems being solved. Since oneM2M architecture is already complex, AI and ML services should be kept as much simple as possible in order to avoid adding complexity to the oneM2M architecture.

8.2.2.3 Fast learning curve

In general, the artificial intelligence community has greater expertise of data analysis, algorithms and mathematics, and less so with Internet of Things and interoperability design approaches. To promote oneM2M in the AI domain, its learning curve for application development should be reduced and simplified.

If oneM2M finds it feasible to reduce or simplify the learning curve of its architecture and make oneM2M more approachable to newcomers, the framework will become more appealing and people will not feel so overwhelmed with what they have to learn/develop to comply with the standard. Regarding AI in particular, the oneM2M community should reflect on its positioning and decide if the framework should implement Machine Learning methods, if it should focus on supplying helper functions as CSFs and tools (brokers) for data collection to support the development AI applications, based on IoT data and devices, or both

8.2.2.4 Supporting material

Solution developers with a focus on AI do not need complete knowledge of oneM2M to apply its specifications to their solutions. However, there is no easy way for developers to explore application possibilities through the existence of a 'quick start' guide, for example. The learning process is lengthy and likely to dissuade many developers. A general recommendation is to leverage the lessons from the team's use-case analysis and PoC development experience to publish supporting materials targeted for IoT system architects and AI developers.

8.2.3 Recommendations for further work

8.2.3.1 Synergies with Release 4 features definition

8.2.3.1.1 Rule-based AI and IoT Data Streams

The earlier analysis of use-cases identified opportunities to superimpose rule-based AI functionality on IoT data streams. This would support monitoring, detection, and the triggering of alerts to applications when anomalous changes were recognized in the state of a connected device. While such monitoring could occur in end-point devices or sensors (AEs), an alternative model is to consolidate such activity (potentially scaling-up across multiple devices and sensors) within the service layer. A topic for future study is to use the <processManagement> resource type that is planned for inclusion in the Rel-4 standard (refer to oneM2M TS-0001 [i.2]). A <processManagement> resource defines a process, consisting of a sequence of states, that a hosting CSE manages on behalf of an AE. This resource type could be combined with the Action Triggering procedure to manage the detection of oneM2M events via the <action> and <dependency> resources (refer to oneM2M TS-0001 [i.2]).

8.2.3.1.2 Semantic Reasoning

Another general-purpose function identified through the use case analysis involves the use of capabilities that support reasoning. These would enable fault-tree types of analysis in the context of explainable AI, for example. The Semantic Reasoning functionality in oneM2M specifications is a candidate to support this form of AI (refer to clause 7.11 of oneM2M TS-0034 [i.5]). It encompasses a <semanticRuleRepository> (refer to oneM2M TS-0001 [i.2]) for grouping together a set of <reasoningRules> (refer to oneM2M TS-0001 [i.2]) and a <reasoningJobInstance> (refer to oneM2M TS-0001 [i.2]) which supports one-time as well as continuous reasoning operations.

8.2.3.1.3 Interworking between oneM2M and NGSI-LD

One potential opportunity for standardization in AI use cases is the interworking between oneM2M and NGSI-LD (ETSI GS CIM 009 [i.7]), to leverage the interoperability with devices and applications from the international standard on the one hand, while providing linked data (entity relationships), property graphs and semantics (exploiting the capabilities offered by JSON-LD). This would not require translation information models from oneM2M to NGSI-LD, but instead to leverage on SAREF ontologies to handle this interwork. The SAREF suite of ontologies, defined under the auspices of the ETSI SmartM2M Technical Committee, provide definition of generic classes that can be mapped to the NGSI-LD cross-domain ontology, but they do also further define domain-specific vocabularies in various domains (city, building, agrifood, etc.), which will allow to establish associations between instances using linked data, a very useful approach for data traceability and correlation, both in the training of AI agents as well as in providing more explainable results from AI use cases.

8.2.3.2 Tools in support of simplicity

The smart city use-cases involving the processing of (video) image data and the identification of faulty sensors based on signal variation patterns illustrate the wide variety of AI/ML tools available to application developers. The deployment of such tools is a case of matching the right AI/ML tool to a given problem, followed by configuration of the tool. There is a balance to be struck in how such tools are integrated into oneM2M standards. It will be important to respect the common services and horizontal architecture characteristics of oneM2M by isolating the common elements of different AI/ML tools for inclusion as common service functions. This will avoid over-complicating oneM2M's common services layer.

Annex A:
Change History

Date	Version	Information about changes
February 2020	0.0.1	Initial draft version (Milestone A) for review at SmartM2M #53
February2020	0.0.2	New version with changes brought during and after STF Meeting #2
February2020	0.0.3	Additional changes to clauses 4 and 5
February 2020	0.1.0	Version uploaded on SmartM2M TC portal as Early Draft
April 2020	0.1.1	First version towards stable draft. Includes material from STF presentation
April 2020	0.1.2	Integration of the changes decided in Meeting #6
April 2020	0.1.3	Integration of the changes
April2020	0.1.4	Version for upload on SmartM2M with revision marks kept.
April 2020	0.2 0	Version for upload on SmartM2M as early draft (same as 0.1.4 without marks)
September 2020	0.2.1	New version to incorporate early results from the PoC
September 2020	0.2.2	Integration of additional material
September 2020	0.2.3	Integration of additional material
September 2020	0.2.4	Pre-final version for sanity check before upload to SmartM2M portal
September 2020	0.2.5	Feedback
September 2020	0.2.6	Feedback
September 2020	0.2.7	Feedback. Identical to final version v 0.9.0
November2020	0.9.1	Update of v0.9.0 with changes
December 2020	0.9.2	Inclusion of contributions after the presentation meeting to oneM2M
December 2020	0.9.3	New contribution on oneM2M and NGSI-LD. Uploaded as final version 1.0
December 2020	1.1.1	ETSI Technical Officer review for ETSI EditHelp pre-processing for publication

History

Document history		
V1.1.1	February 2021	Publication