The Standards People

oneM2M Technical Plenary #47

AI/ML for IoT: Enriching oneM2M with generic AI capabilities

A Proof-of-Concept

Presented by: **Ken Figueredo** For: **ETSI Specialist Task Force 584**

3 Nov. 2020

Agenda

- Project introduction and context (15 minutes)
- Use cases implemented by Sensinov (30 minutes)
- Use case implemented by Ubiwhere (15 minutes)
- Concluding discussion and avenues for future work (30 minutes)

Project Introduction & Context

Artificial Intelligence for IoT Systems

Objectives

- Provide an initially validated architecture that describes how IoT systems can make use of Artificial Intelligence (AI) and Machine Learning (ML) for the management and interpretation of data from IoT devices

- Identify the requirements to extend/adapt oneM2M to improve the integration of AI/ML techniques and ensure that the associated management of data is properly handled by the IoT Service Layer

Project Team and Approach

Multi-disciplinary and International Project Team

- Mahdi Ben Alaya, Sensinov
- Ricardo Vitorino, Ubiwhere
- João Garcia, Ubiwhere
- Emmanuel Darmois, CommLedge
- Ken Figueredo, More with Mobile

PROJECT PLAN & DELIVERABLES

Identify 'AI for IoT' Architecture Evolutions	Proof of Concept Implementations	Dissemination toward IoT and oneM2M community
ETSI TR 103 674: SmartM2M: Artificial Intelligence and the oneM2M architecture	**ETSI TR 103 675: SmartM2M: AI for IoT - Proof of Concepts and Open-source Code**	**Presentation at oneM2M Technical Plenary #47**

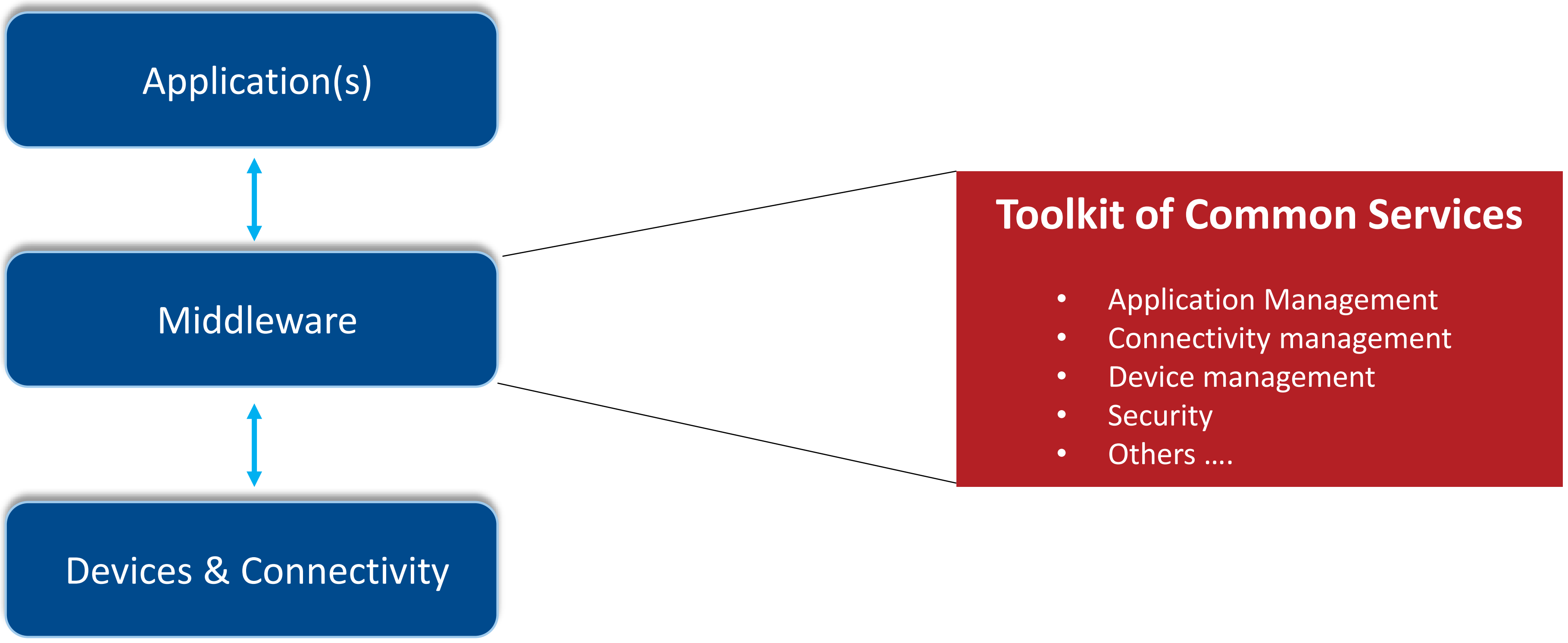
Analysis of representative use-cases

AI/ML USE-CASES: POINT-SOLUTION & HORIZONTAL CAPABILITIES

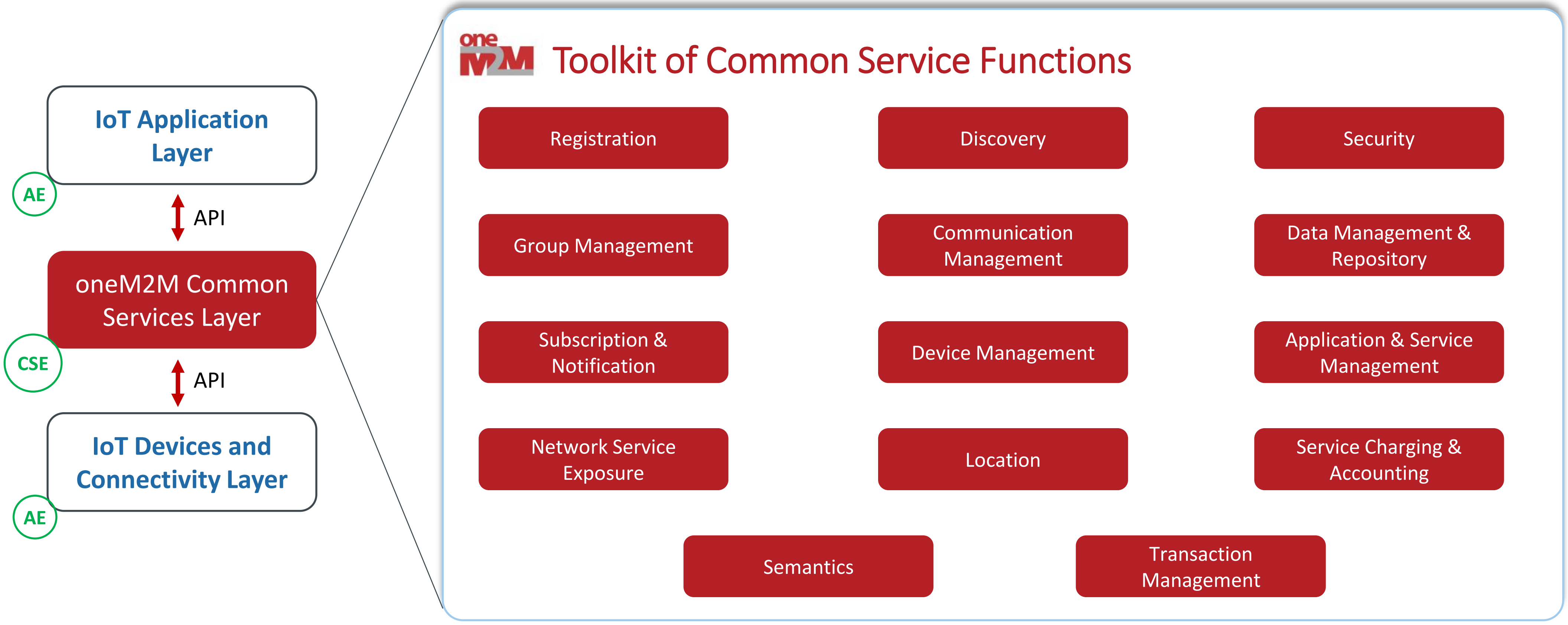
ID	Use-Case
1	Fault management and isolation for IoT field devices
2	Detection of patterns in video streams
3	Language-based pattern recognition in social media/crowdsourced data
4	Supervisory road-network traffic management for journey planning or to manage transportation logistics
5	Autonomic Management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

SOURCE: ETSI TR 103 674: SmartM2M: Artificial Intelligence and the oneM2M architecture (to be published)

Definitions: A generic IoT system



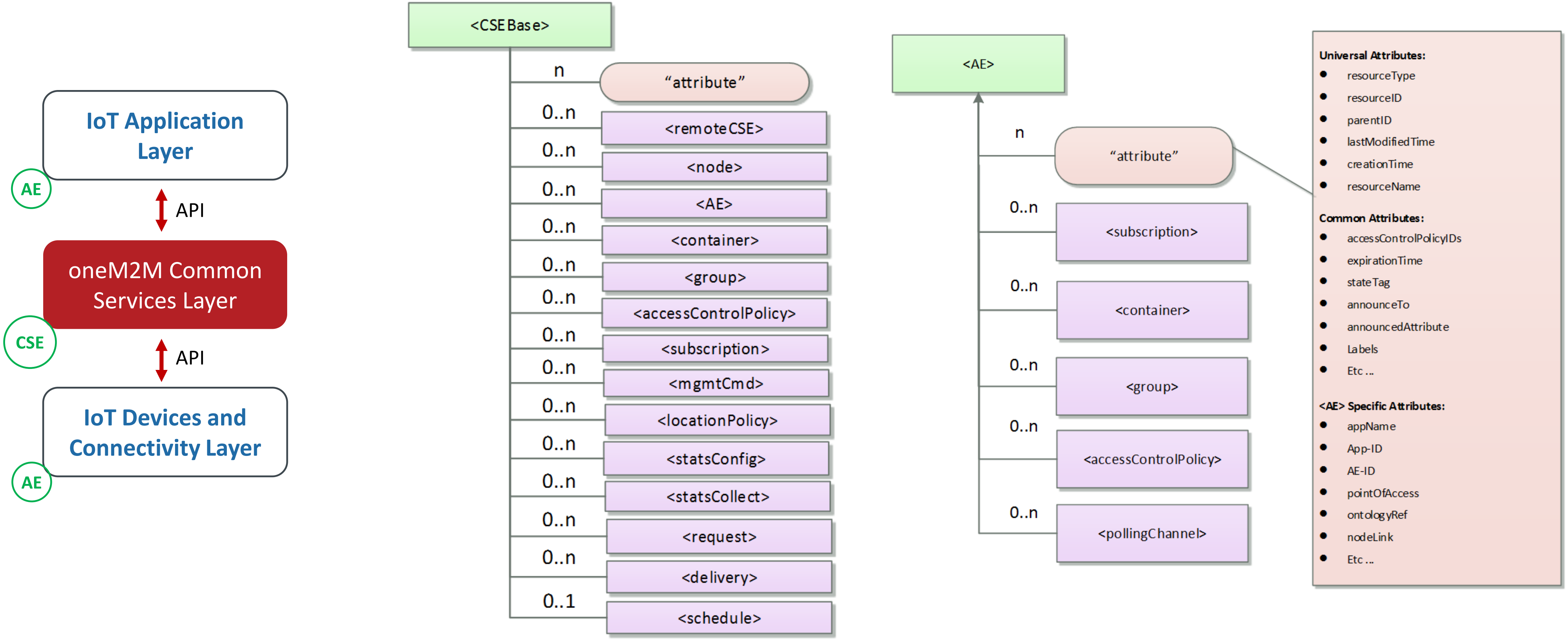
oneM2M Framework: A single API for a toolkit of common service functions



AE In oneM2M terminology, an AE represents an Application Entity

CSE In oneM2M terminology, an CSE represents an Common Services Entity

oneM2M Framework: All entities in a oneM2M system are represented as addressable resources in the CSE



SOURCE: https://www.onem2m.org/getting-started/onem2m-overview/application-program-interfaces-api/onem2m-resources

Framework for AI/ML in IoT

Knowledge Rules-based AI

Reasoning
- Rule-based
- Pattern recognition

Decision Making
- Recommendations
- Automated actions

Supervision
- Meta reasoning
- Federated reasoning

Explainable AI
- Causal inferencing
- Fault-tree analysis

Signal Processing / Machine Learning

Data Processing
- Filter
- Normalise or Contextualise
- Feature extraction
- Fusion/Machine Learning (combine streams)
- Data mining/ unsupervised learning

Knowledge Representation
- Pattern recognition/ Classification
- Symbolic mapping/ representation
- Data mining/ supervised learning

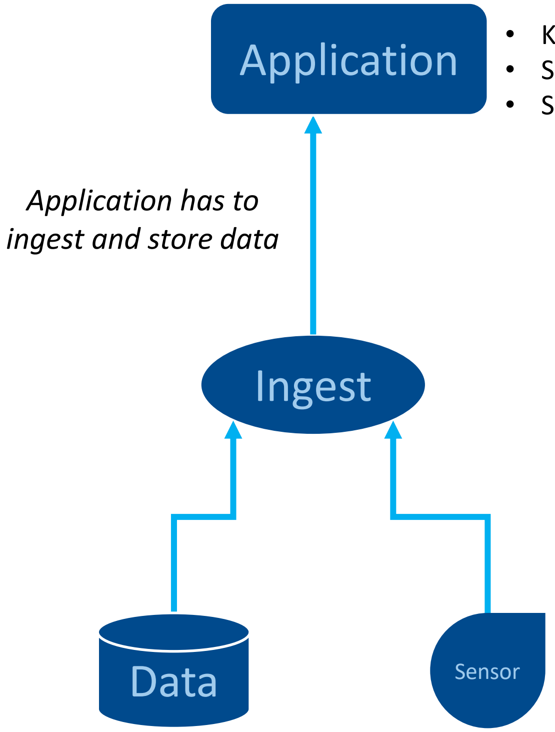
AI/ML in the Application Layer

Device/Data Management

Data Sourcing
- Device Management
- Communications Management
- Data management
- Registration & Identity Management
- Authentication
- Other common services

Service Layer

AI/ML for IoT Challenge: Traditional Approach



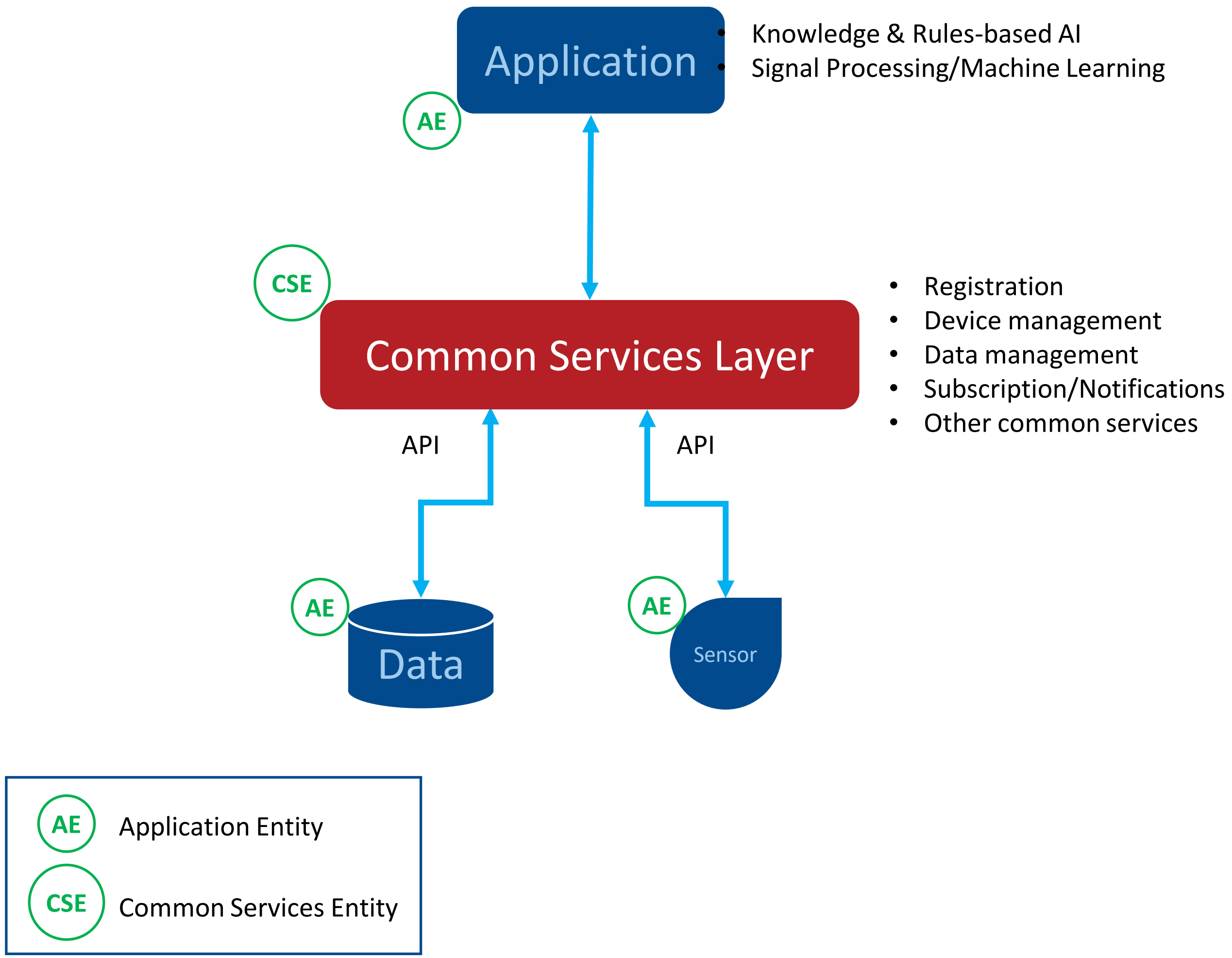
- Knowledge & Rules-based AI
- Signal Processing/Machine Learning
- Service layer functions

× *Concentrates all intelligence in the Application layer*

× *Developers recreate basic capabilities for each application they build*

× *Limitations for reuse and replicability*

AI/ML for IoT Challenge: Scalable & Developer-friendly Approach



- ✓ *Create a bi-directional abstraction layer between Applications and the underlying technologies for data and sensor management*
- ✓ *Developers use a standard API in combination with a Subscription/Notification service to access data when notified of changes*

Two implications for Common Services Layer capabilities

1. Convert AI/ML capabilities into "as-a-service" capabilities in the Common Services Layer

Knowledge Rules-based AI

Signal Processing / Machine Learning

Device/Data Management

Two implications for Common Services Layer capabilities

Knowledge Rules-based AI

Signal Processing / Machine Learning

Device/Data Management

1. Convert AI/ML capabilities into "as-a-service" capabilities in the Common Services Layer

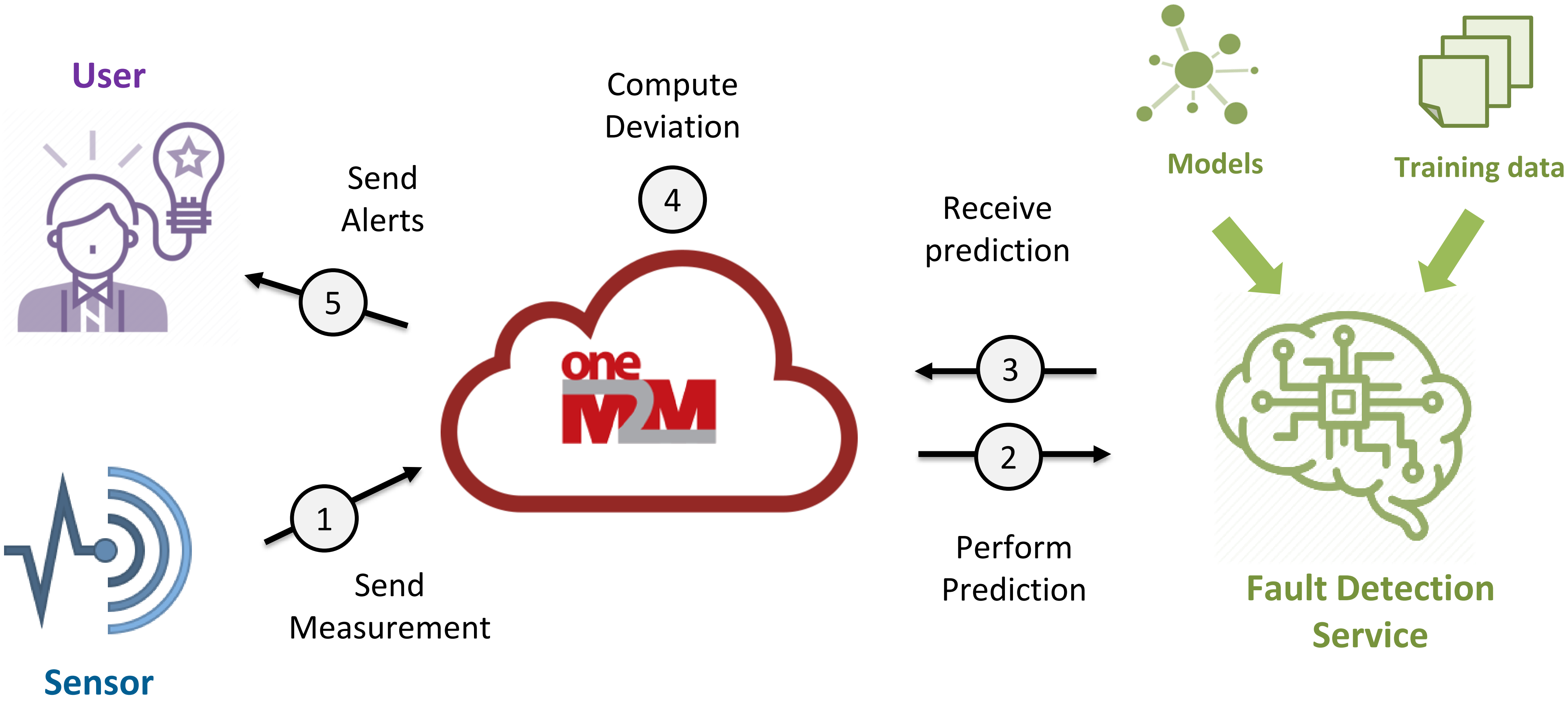
2. Feed CSL information to applications in order to improve AI/ML performance

Three use-cases selected Proof of Concept testing

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

Use Case 1 - Fault Management and Isolation for IoT Field Devices (Sensinov)

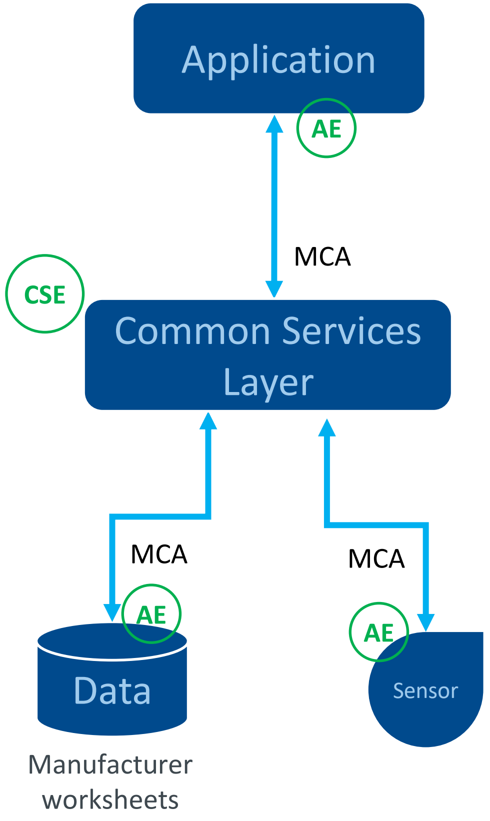
Use Case 1: Fault management and isolation for IoT field devices
Overview



Architecture

- Fault detection aims to identify defective states and conditions within IoT systems, based on measurements from field devices
- In this use case, an IoT module will be prototyped for fault detection and isolation in a smart building environment using both a rule-based and a self-learning fault detection algorithms
- The main goal is to extend oneM2M with fault detections capability to make it possible for oneM2M developers and users to detect malfunctions in real time, as soon and as surely as possible.

Fault Management PoC Architecture



- Configure fault detection
- Receive notification when anomalies are detected

- Accept Fault detection configuration parameters from AEs
- Fault detection service: Rule-based/ML-based
- Send notification to AEs

- Send measurements

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

Impact on oneM2M

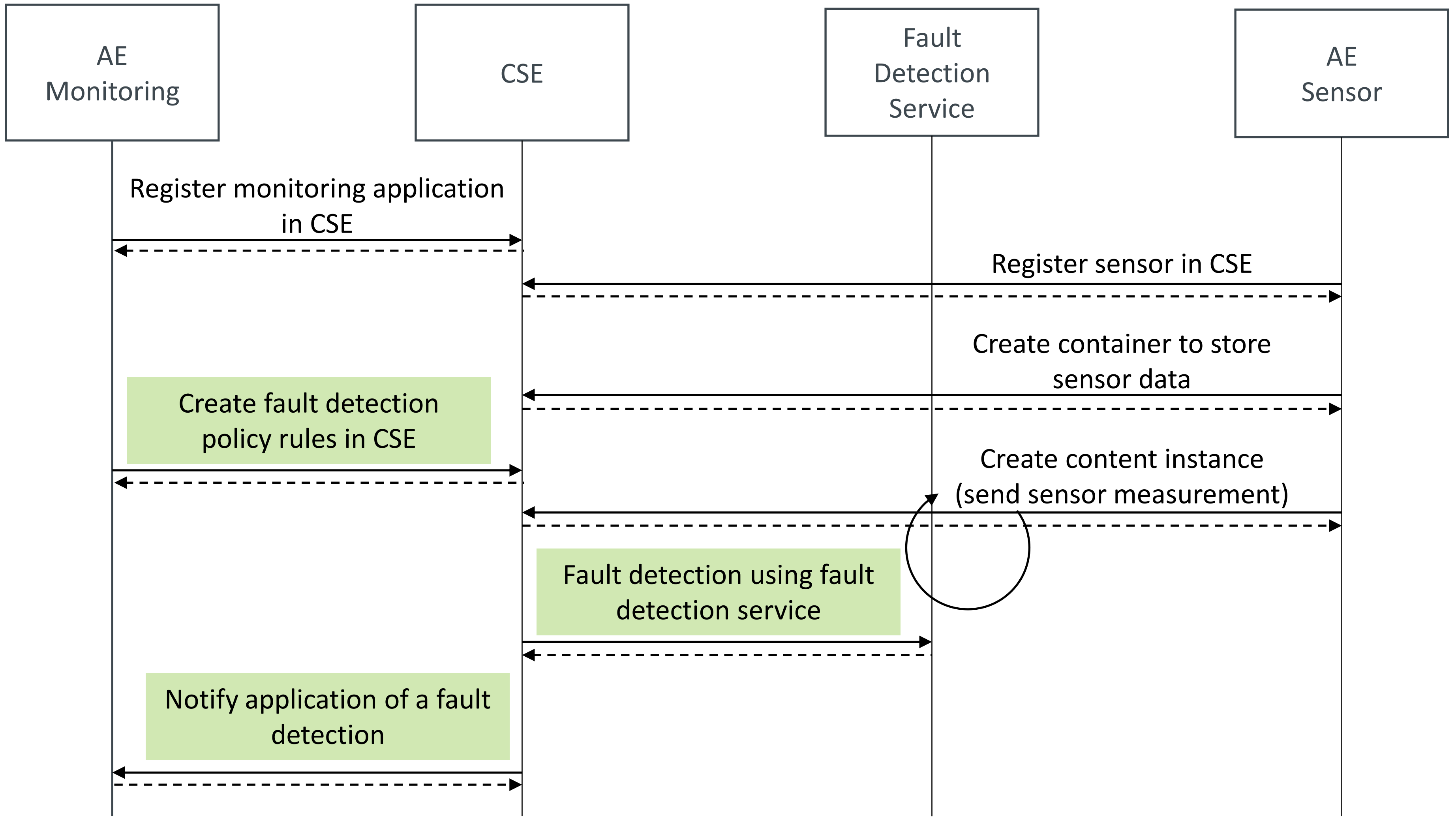
Common Service Function modifications

- Data management: extend oneM2M resources (e.g. containers, flexContainers, 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.

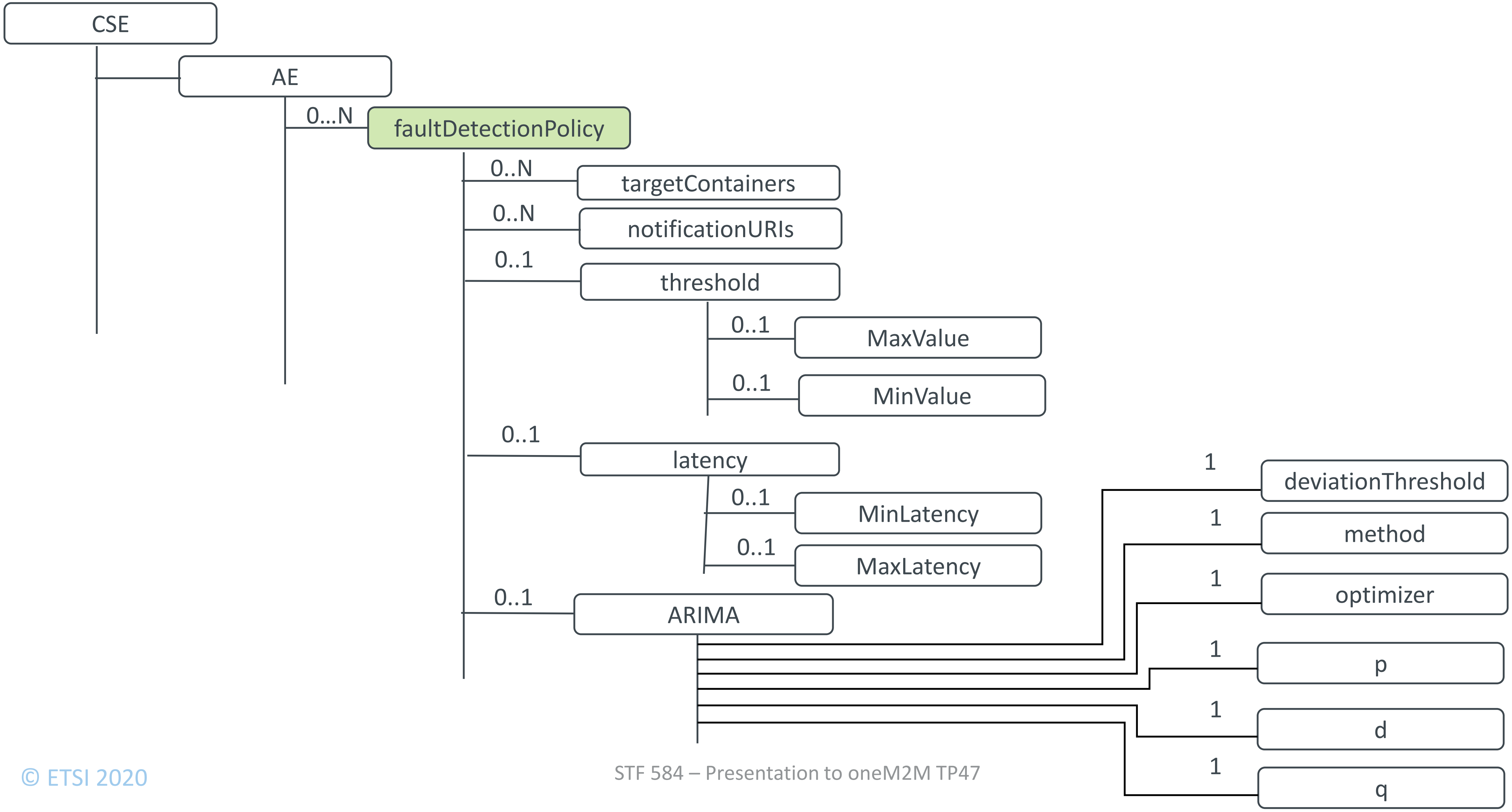
New AI/ML CSF requirements

- 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

Use Case 1: Fault management and isolation for IoT field devices
Call flow



Resource tree

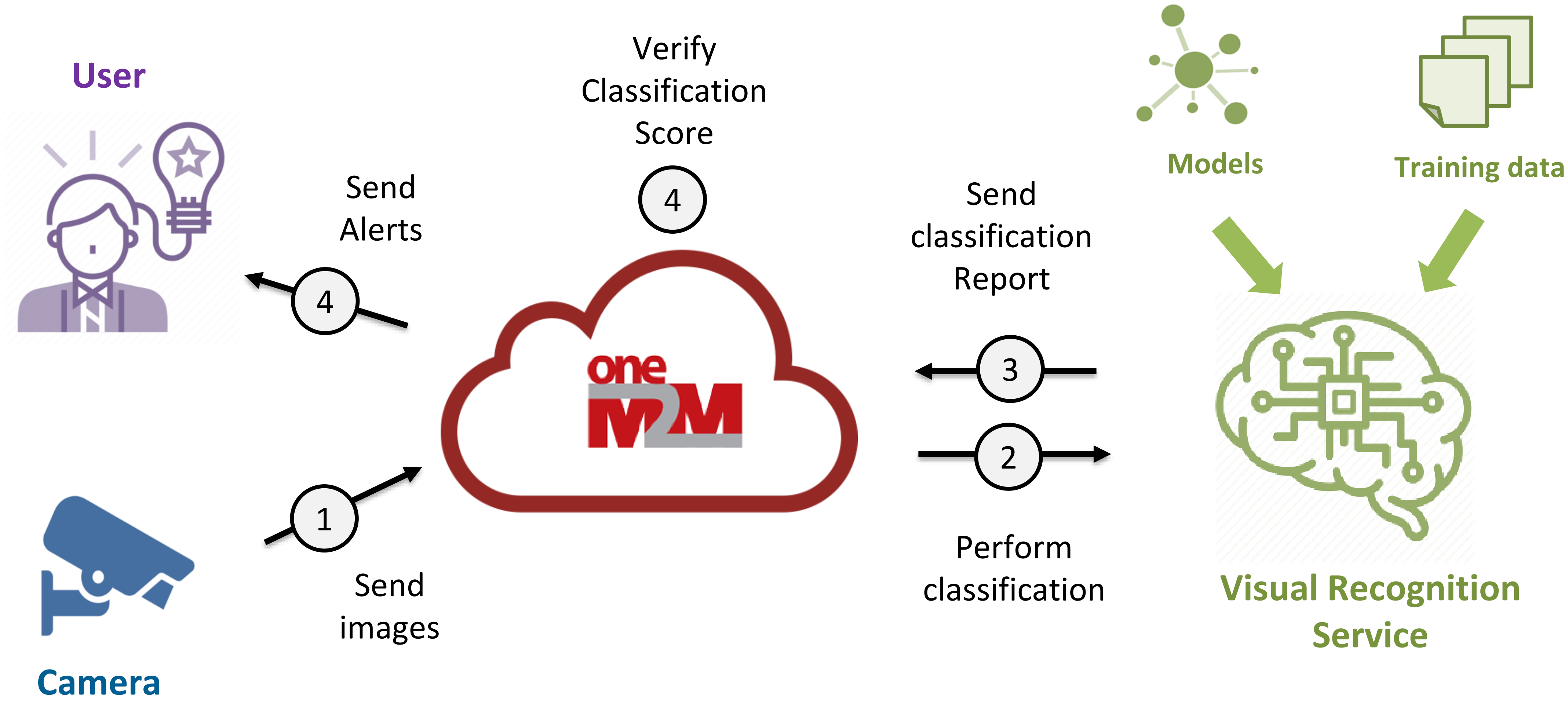


Demonstration

Demonstration UC1

Use Case 2 – Detection of Patterns in Video Streams (Sensinov)

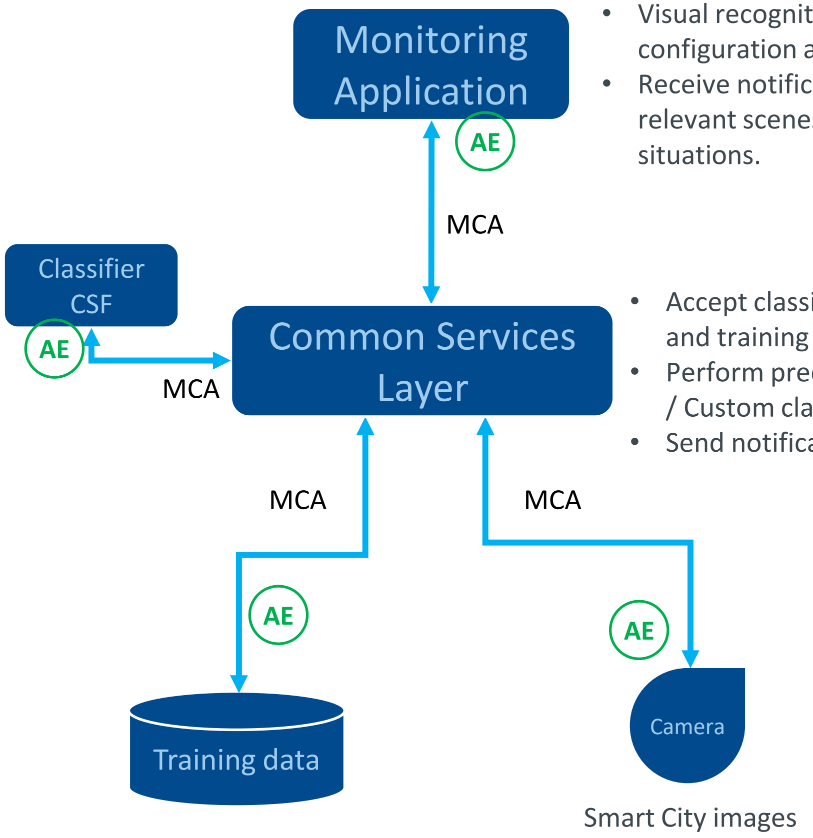
Use Case 2: Detection of patterns in video streams
Overview



Architecture

- 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 performs image classification using machine learning and trained data. A camera periodically reads images from a data store and pushes them to the oneM2M platform and receive notifications when trained categories are recognized.
- The main goal is to extend the oneM2M architecture with visual recognition capabilities to make it possible for oneM2M developers to gain high-level understanding from digital images through the construction of explicit, meaningful descriptions of physical objects, and scenes and use them to make relevant decisions.

Video Pattern Detection PoC Architecture



- Visual recognition configuration and training
- Receive notification related to relevant scenes, objects, and situations.

- Accept classification configuration and training data from AEs
- Perform predefined-classification / Custom classification
- Send notification to AEs

Use Case 2: Detection of patterns in video streams
Impact on oneM2M

Common Service Function modifications

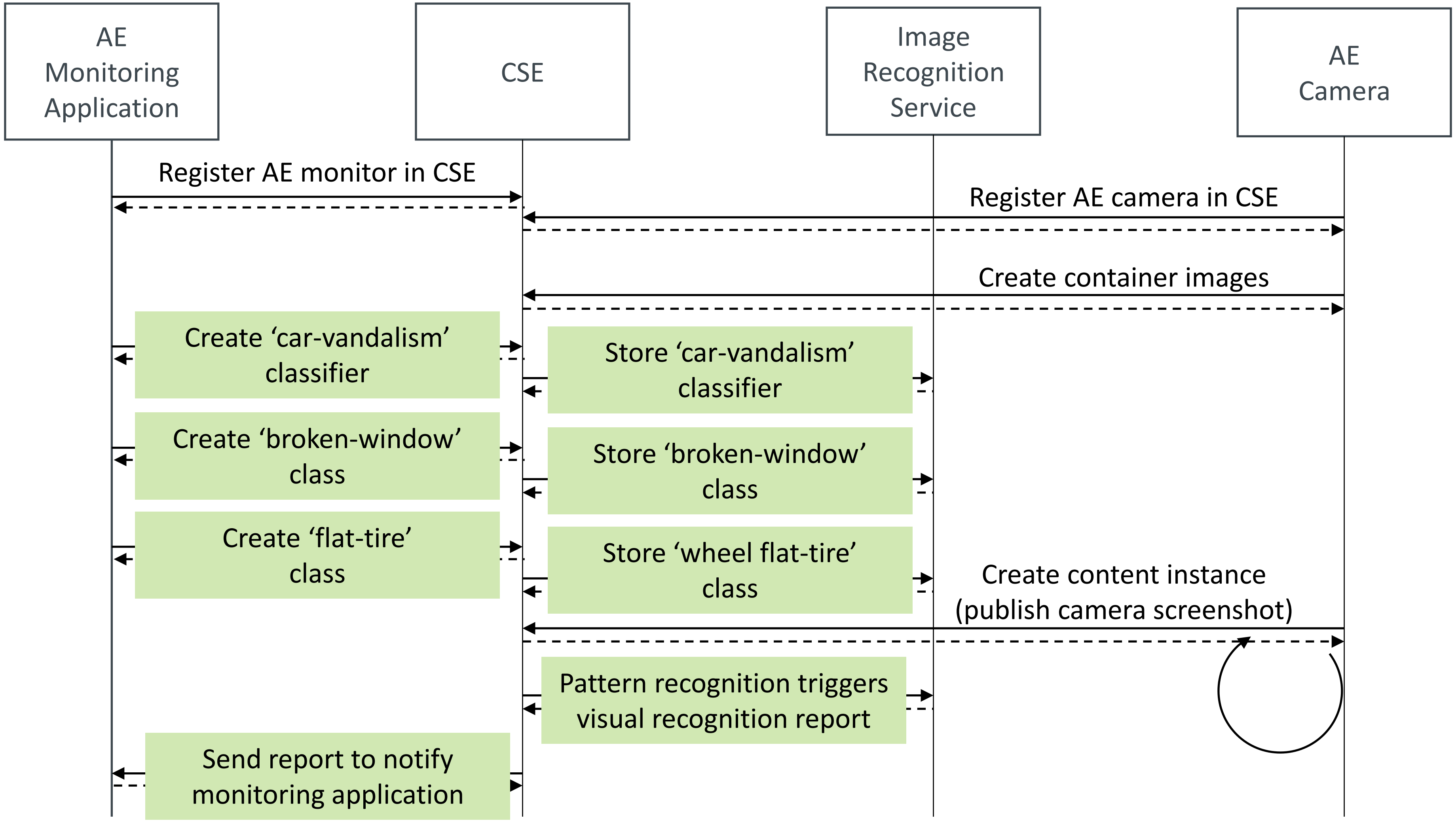
- Data management: use existing oneM2M resources (e.g. containers, flexContainers, etc.) or create new ones to store training images, real images coming from cameras, and the result of the classification.
- Application and Service Layer Management: Extend the oneM2M MCA interface to support the configuration 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 AI/ML CSF Requirements

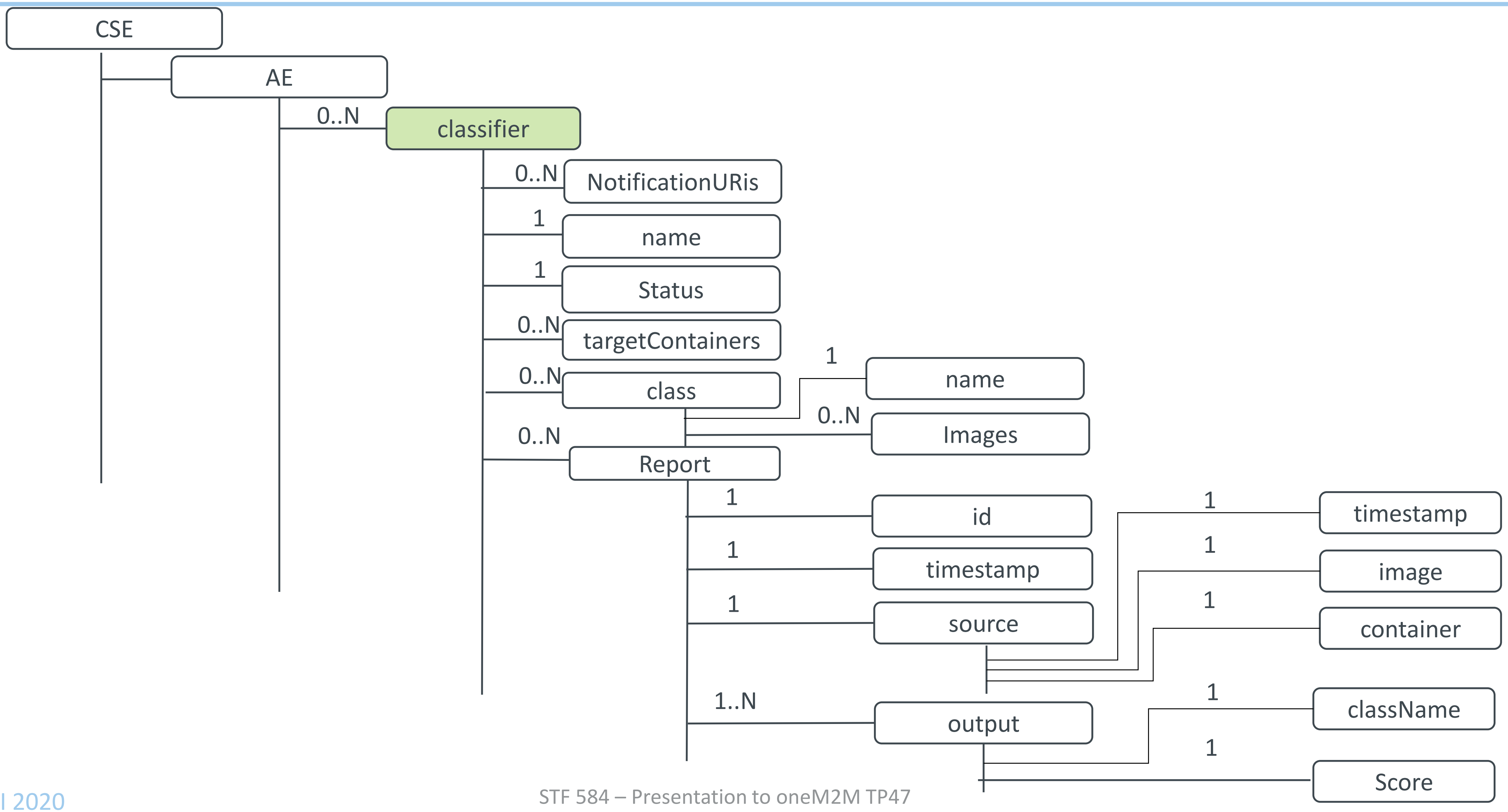
- Predefined-classifier CSF. The CSF comes with a predefined and pretrained classifier for Object detection, Object tracking, Semantic Segmentation, Instance Segmentation, etc.).
- Custom classifier CSF image classification CSF. Allow an application to create its own classifier and train it to implement specific visual recognition use cases.

Use Case 2: Detection of patterns in video streams

Call flow



Use Case 2: Detection of patterns in video streams
Resource tree



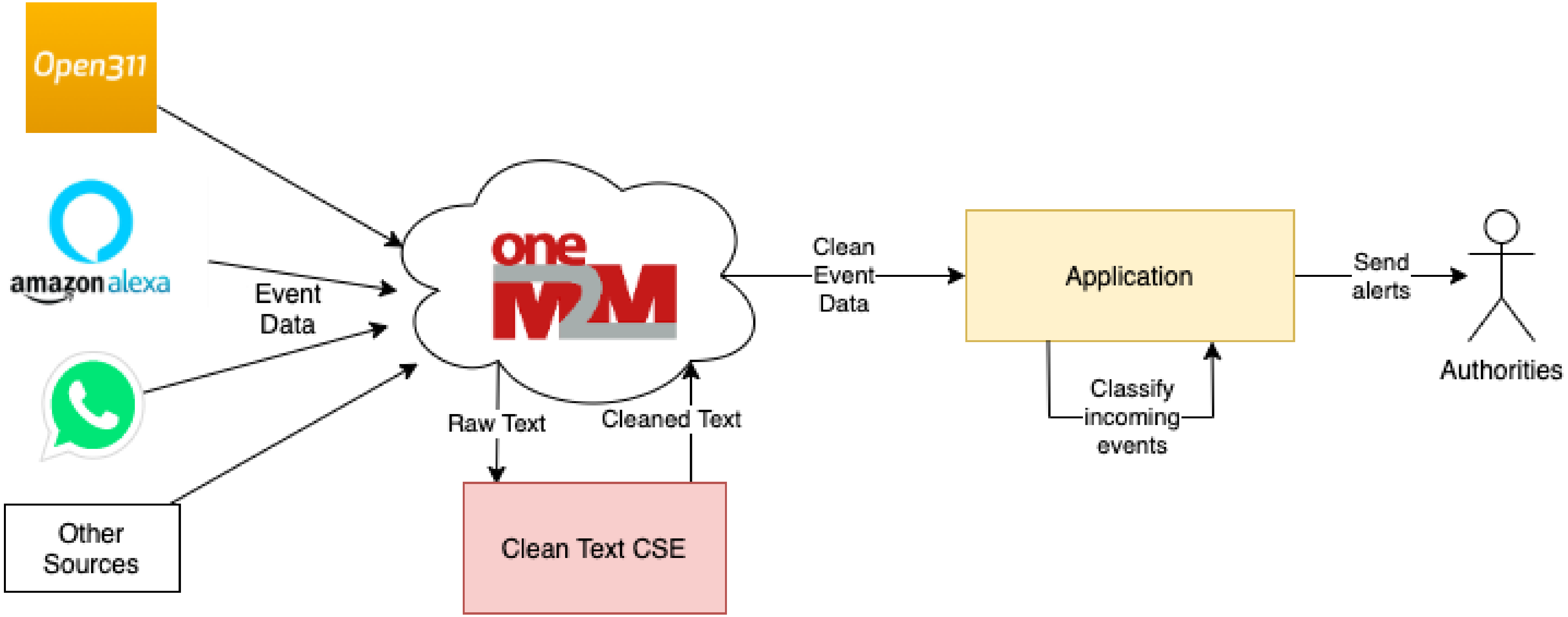
Demonstration

Demonstration UC2

Use Case 3 – Language-based Sentiment Classification (Ubiwhere)

Use Case 3: Language Based sentiment Classification
Overview

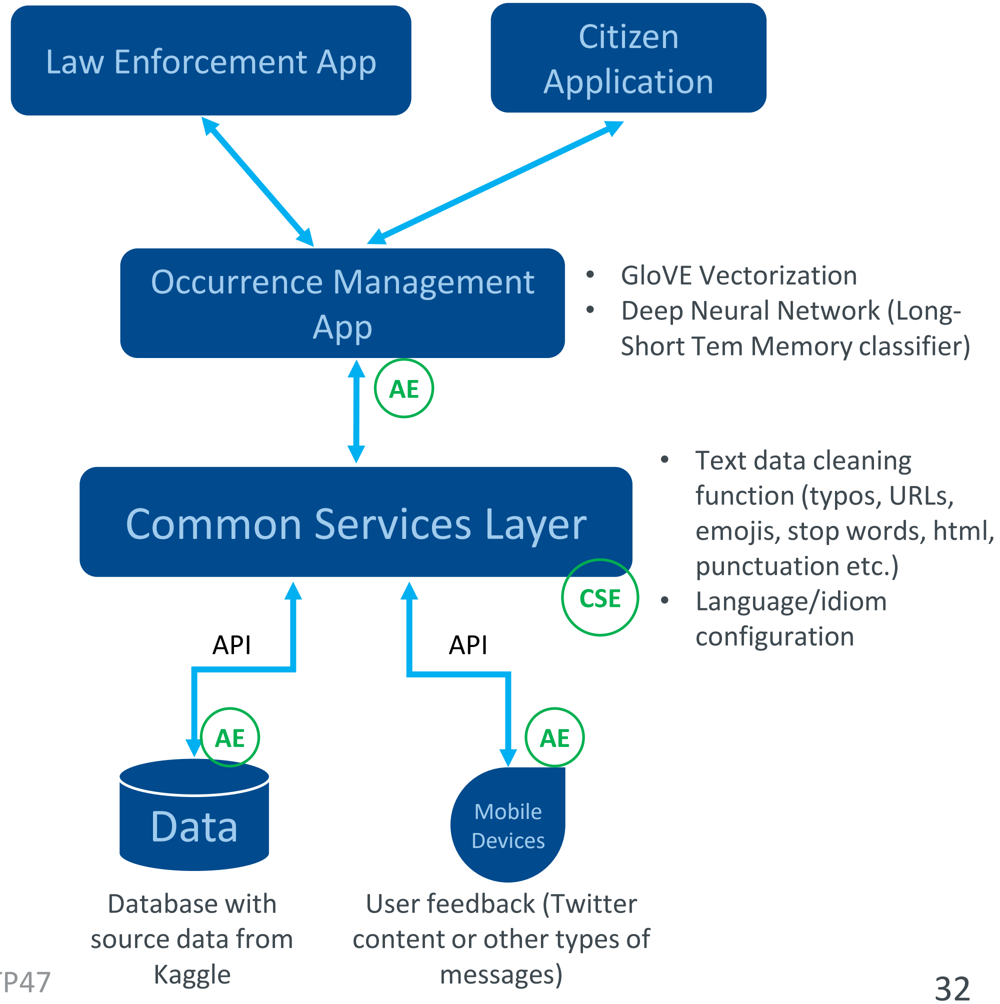
Prior Experimentation Motivated the Use Case 3 PoC



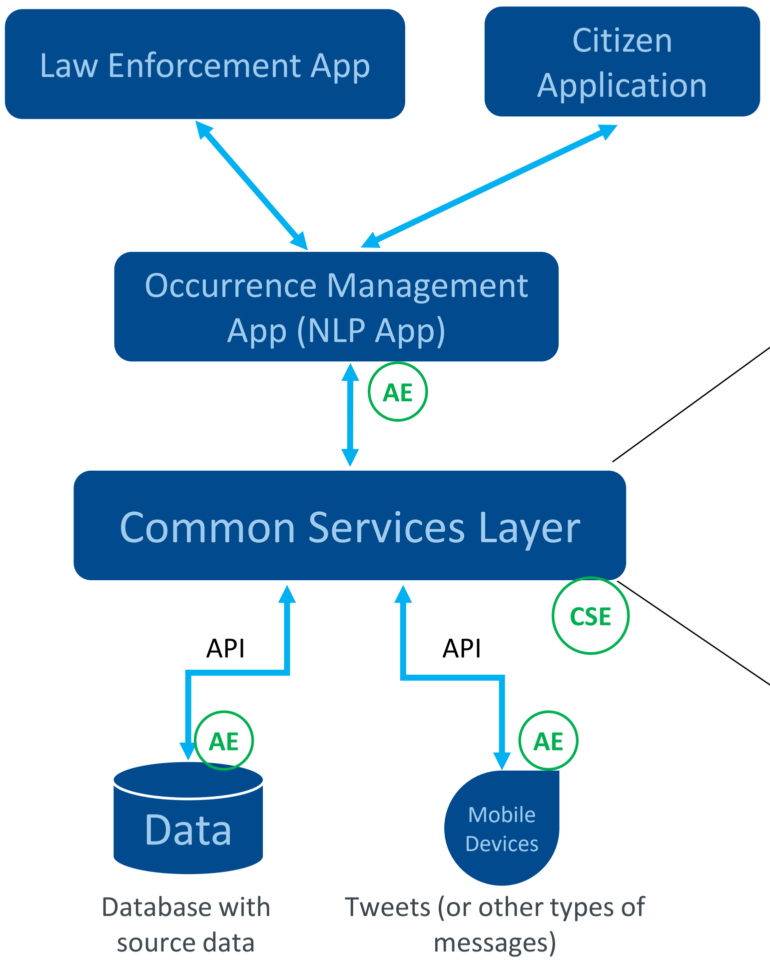
Use Case 3: Language Based sentiment Classification
Architecture

- IoT data can take many forms including numeric and textual.
- The aim of this PoC is to handle text-format data in a smart cities context.
- The use case involves the analysis of crowdsourced text data (e.g. from social networks or specific mobile applications) to detect the occurrence and location of disasters.

Language-based Classification PoC Architecture



Use Case 3: Language Based sentiment Classification
Impact on oneM2M



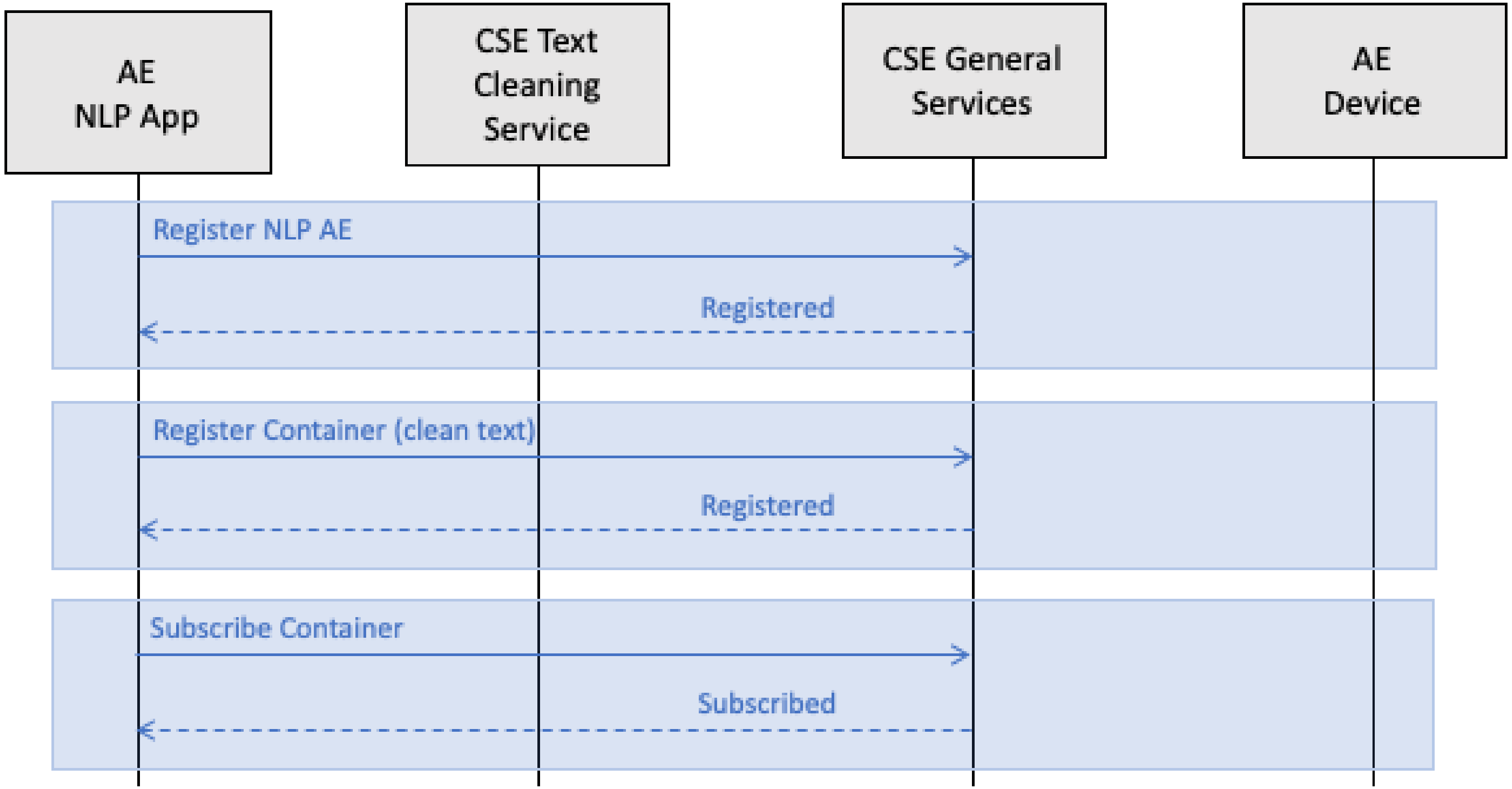
COMMON SERVICE FUNCTION MODIFICATIONS

- 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
- Subscription and Notification: Notify applications when content is published

NEW AI/ML CSF REQUIREMENTS

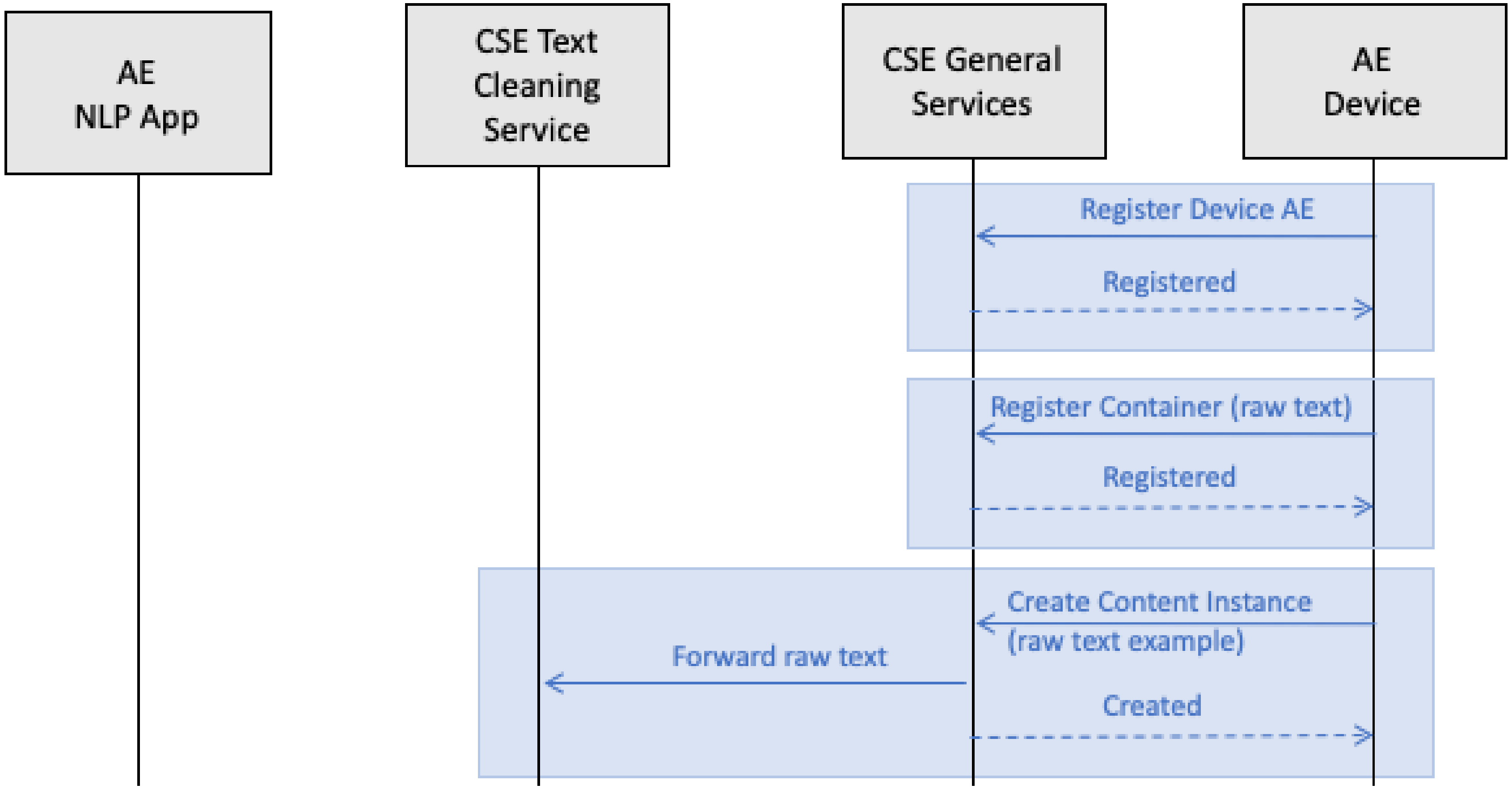
- Feature extraction through common NLP tasks such as Tokenization, Part-of-Speech Tagging, Name-entity recognition, Topic modeling, Speech recognition

Use Case 3: Language Based sentiment Classification
Call flow (1/3)

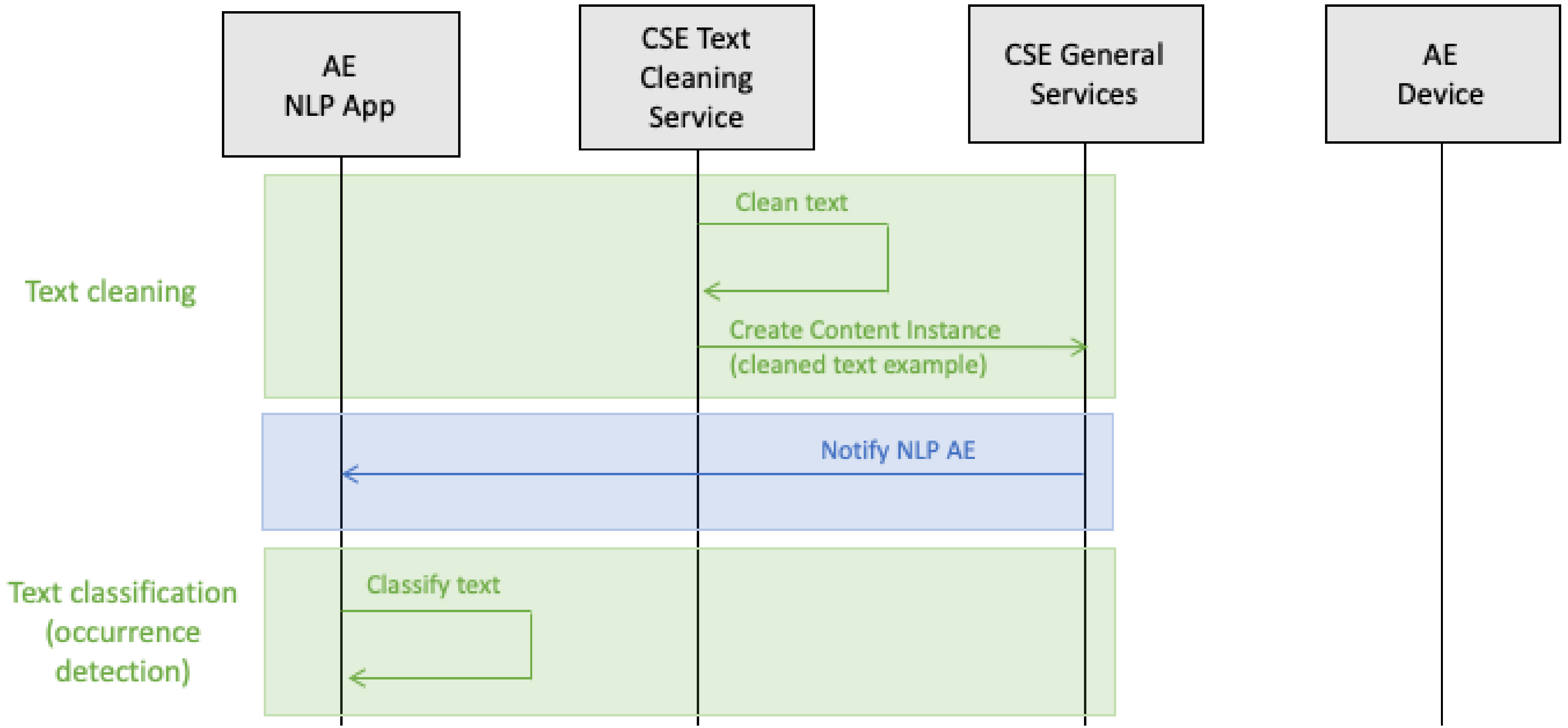


Use Case 3: Language Based sentiment Classification

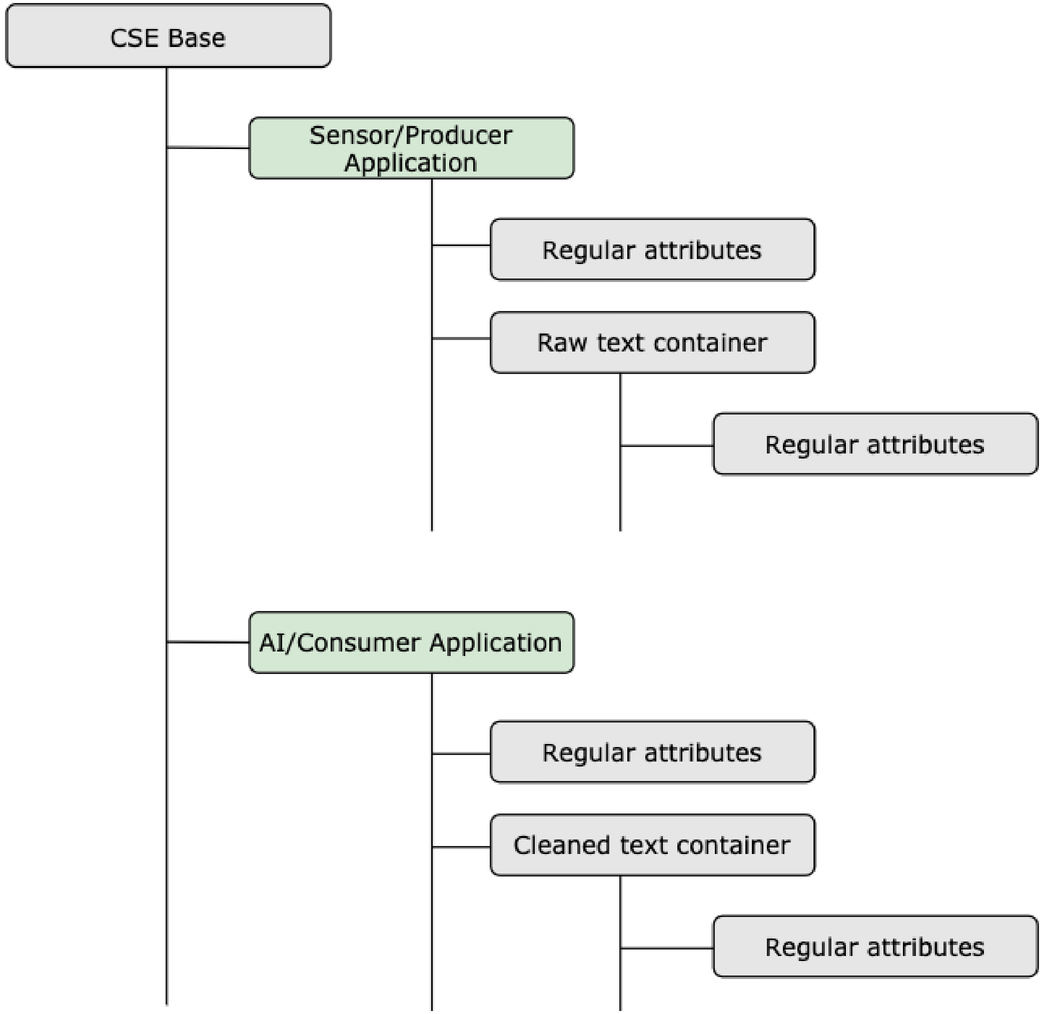
Call flow (2/3)



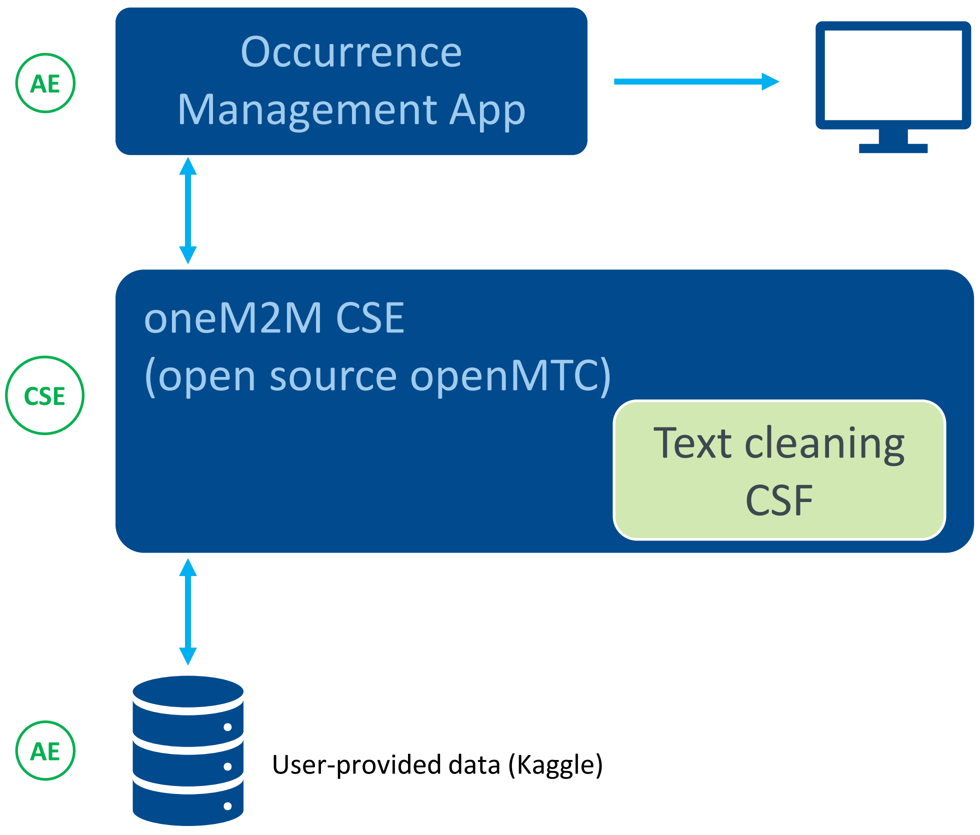
Use Case 3: Language Based sentiment Classification
Call flow (3/3)



Use Case 3: Language Based sentiment Classification
Resource tree



Use Case 3: Language Based sentiment Classification
Test procedure and performance results



NOTE: OpenMTC source - http://www.openmtc.org/

1. Train the NLP Classifier Model in the Occurrence Management App (NLP)
2. Start up oneM2M gateway (OpenMTC)
3. Start up the NLP AE
4. Start up the Device AE

We will see one gateway logging the application, container and content instances being registered and the upper AE (NLP) being notified of content generated by the lower AE (Device).

Use Case 3: Language Based sentiment Classification
Demonstration

Demonstration UC3

Project Findings, Q&A and Next Steps

Project Findings

Lessons learned from PoC testing

- oneM2M's architectural framework is well suited to separate 'decision making' from routine or repetitive 'housekeeping' forms of AI/ML in IoT systems
- Easier to extend CSF capabilities using a micro-services platform architecture compared to a monolithic one

Architectural dependencies that make it 'easy'

- oneM2M's framework, APIs and resource model provide foundations for extensible, 'AI-as-a-Service' tools in the service layer of the IoT solution stack
- Subscription and notification CSFs can enable lighter-weight and operationally efficient AI/ML methods in the applications layer

Discussion and next steps

- Audience questions
  - ....?

- Ideas from project team
  - What approach should oneM2M take to standardize AI/ML in IoT through work items that build on recent data-management initiatives?
  - Retain focus on oneM2M's horizontal approach to standardize general-purpose CSFs
  - Keep AI/ML services as simple as possible to avoid adding complexity to oneM2M
  - Provide educational information and tools that will help (non-IoT-expert) AI/ML developers to experiment with oneM2M

Project deliverables to be published by end-2020

- ETSI TR 103 674: SmartM2M: Artificial Intelligence and the oneM2M architecture
  - Will be publicly available before the end of 2020
- ETSI TR 103 675: SmartM2M: AI for IoT - Proof of Concepts and Open-source Code
  - Will be publicly available this week
- Open source implementations on ETSI GitHub portal
  - Available to ETSI members at https://forge.etsi.org/rep/stf584

Thank you for your attention

STF584: https://portal.etsi.org/STF/STFs/STFHomePages/STF584

Contact Details:

Mahdi Ben Alaya, Sensinov
benalaya@sensinov.com
+33 6 71 93 90 45

Emmanuel Darmois, CommLedge
emmanuel.darmois@commledge.com
+33 6 32 51 53 93

Ken Figueredo
ken@more-with-mobile.com
+1 202 262 7344

Ricardo Vitorino, Ubiwhere
rvitorino@ubiwhere.com
+351 913 850 631