Open Source and standardized platform for Internet of Things and M2M systems

Nicolas Seydoux

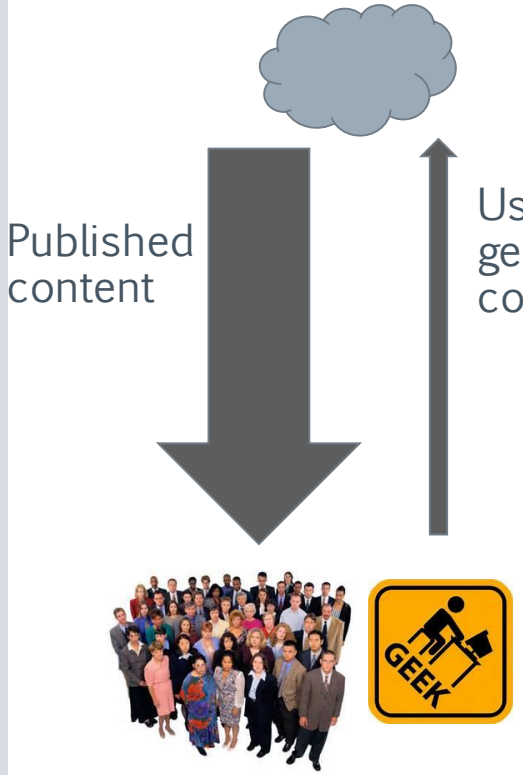
Nicolas.seydoux@laas.fr

Outline

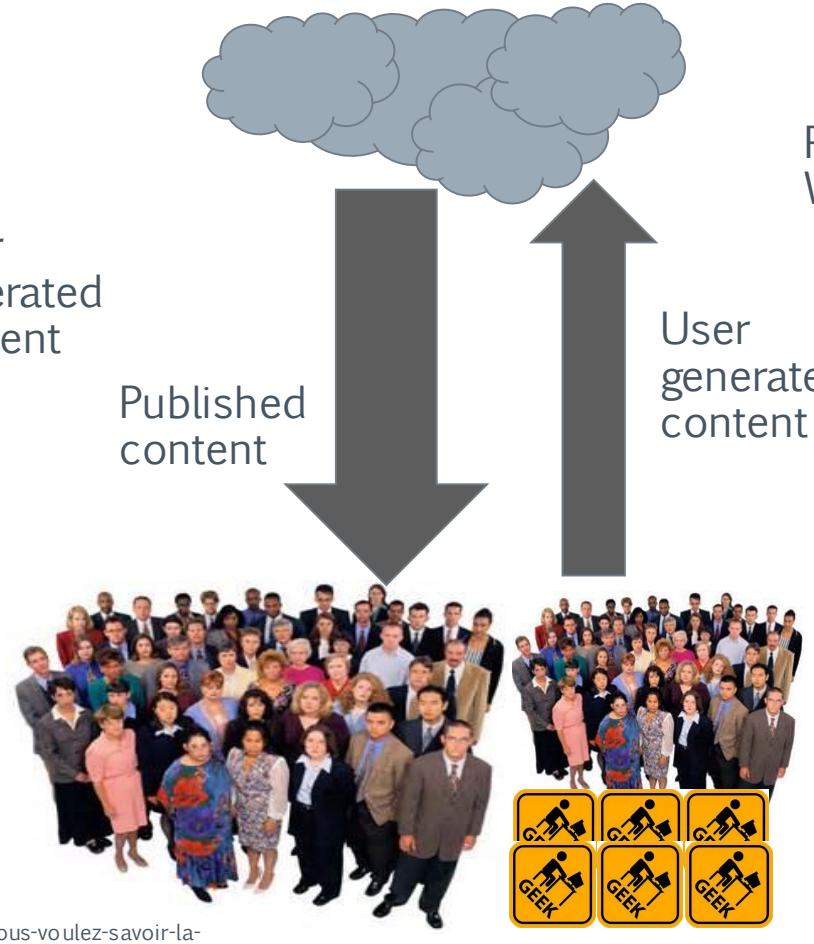
- Internet of things: definition and use cases
- A standard for IoT: OneM2M
- OM2M – an eclipse opensource project
- Focus on applications and live demo

Evolution of WEB

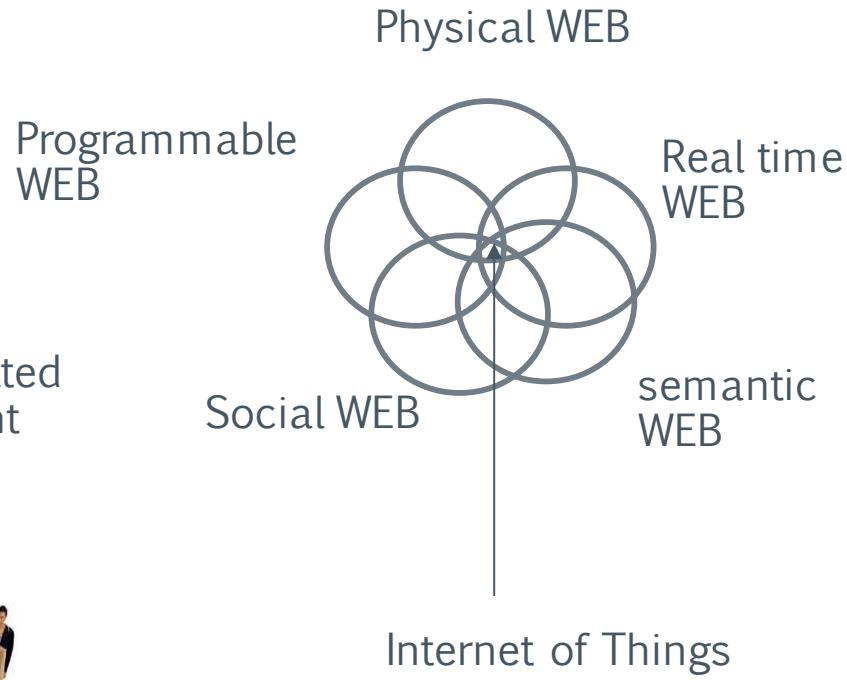
1996
WEB 1.0
=> The mostly read only WEB with 250000 sites



2006
WEB 2.0
=> read-write Web with 80000000 sites



Few years later ...
WEB 3.0 / Web of things[2]



Ref: http://www.webschool-tours.fr/web-1-0-web-2-0-vous-voulez-savoir-la-difference/

2-http://www.webofthings.org/2011/02/04/lift11-talk-transcript/

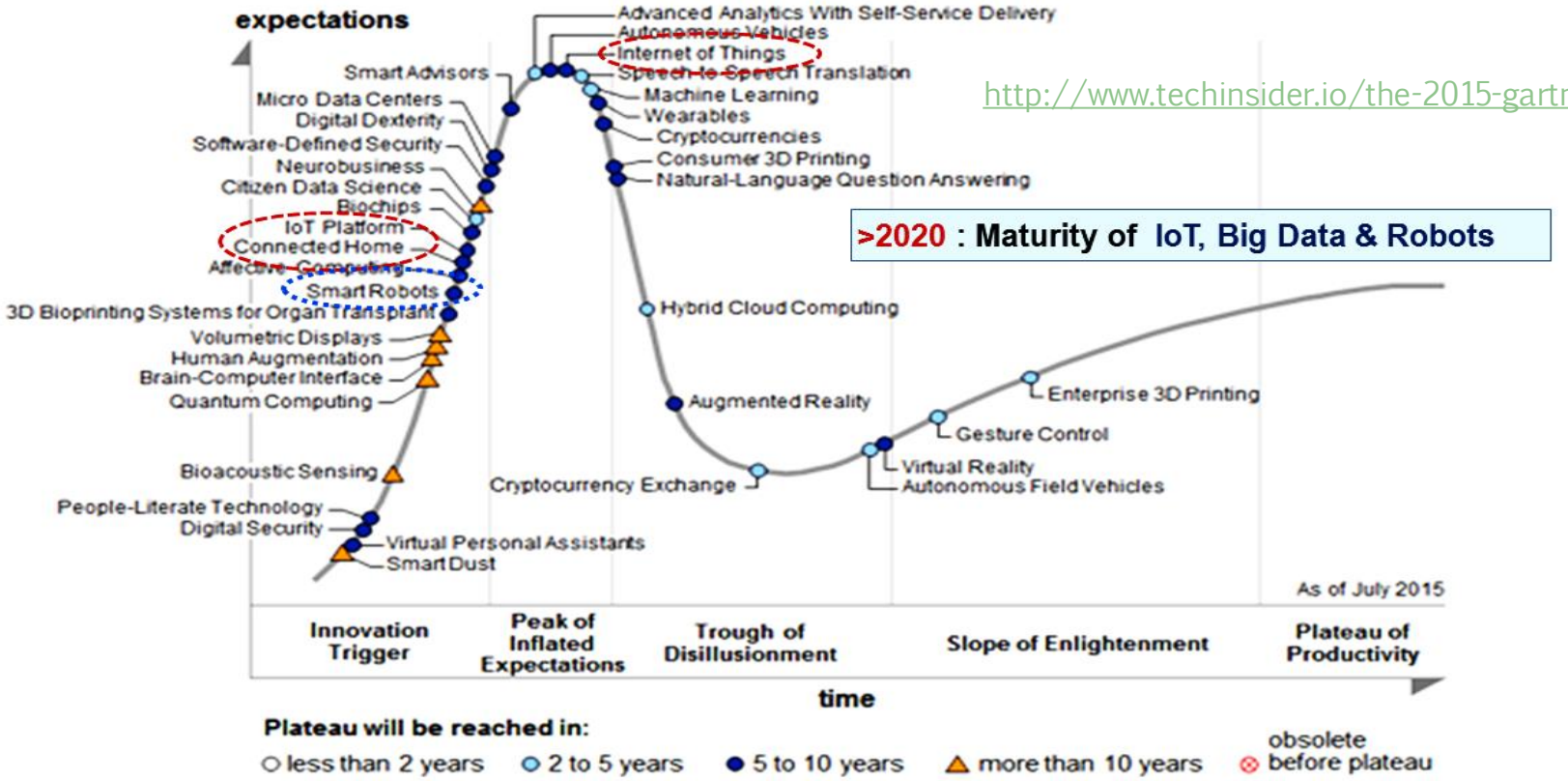
Definition of Internet of Things

- from UIT: A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.[1]
- Key points:
  - A network of network
  - Use of wireless networks
  - Ability of Physical and virtual entities to communicate
  - An address for contents but also for physical entities

1- UIT-T Y.2060 (06/2012)

Potential of Internet of Things

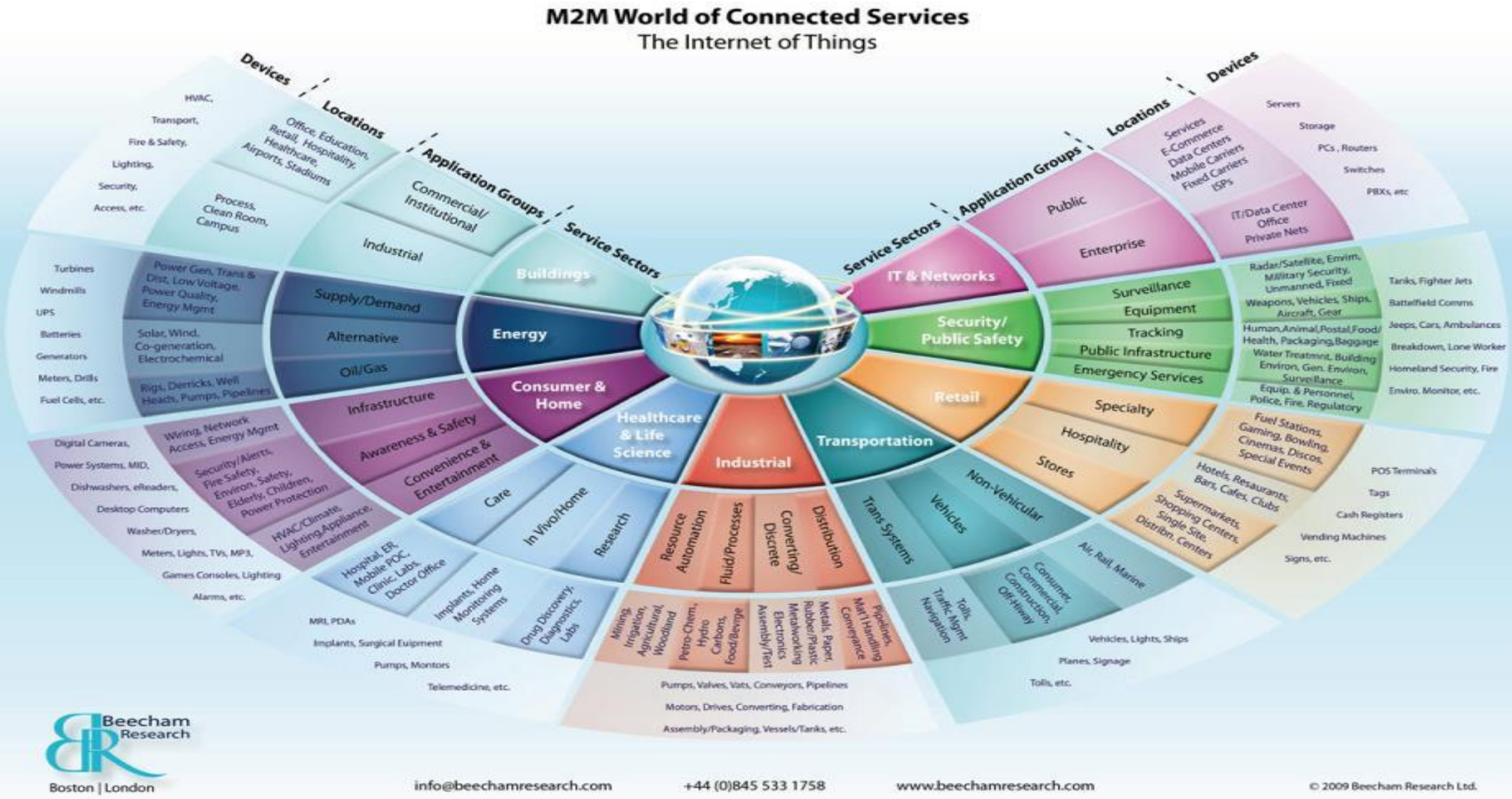
- Maturity of the market in 2020



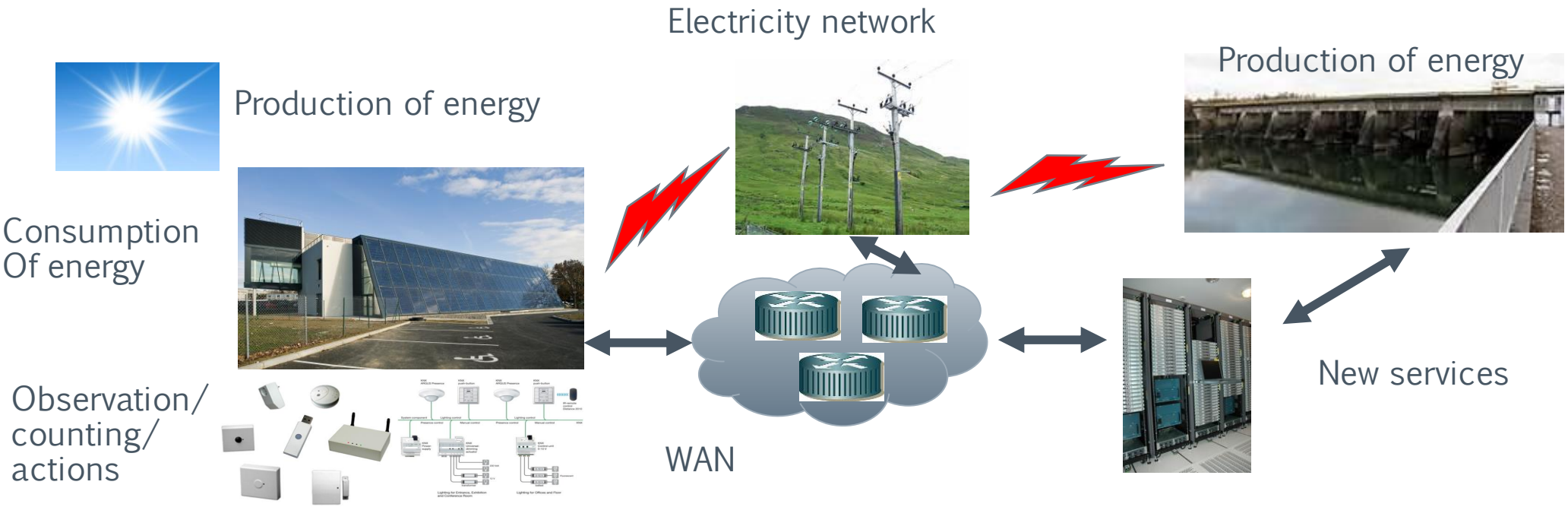
http://www.techinsider.io/the-2015-gartner-hype-cycle-chart-2015-8

- Different analysis (Gartner, ABI Research, Cisco)
  - 2014: between 4 to 15 billion of objects => between 25 to 49 billions en 2020
  - Market : between 1,7 to 13 trillion € in 2020

Application domains

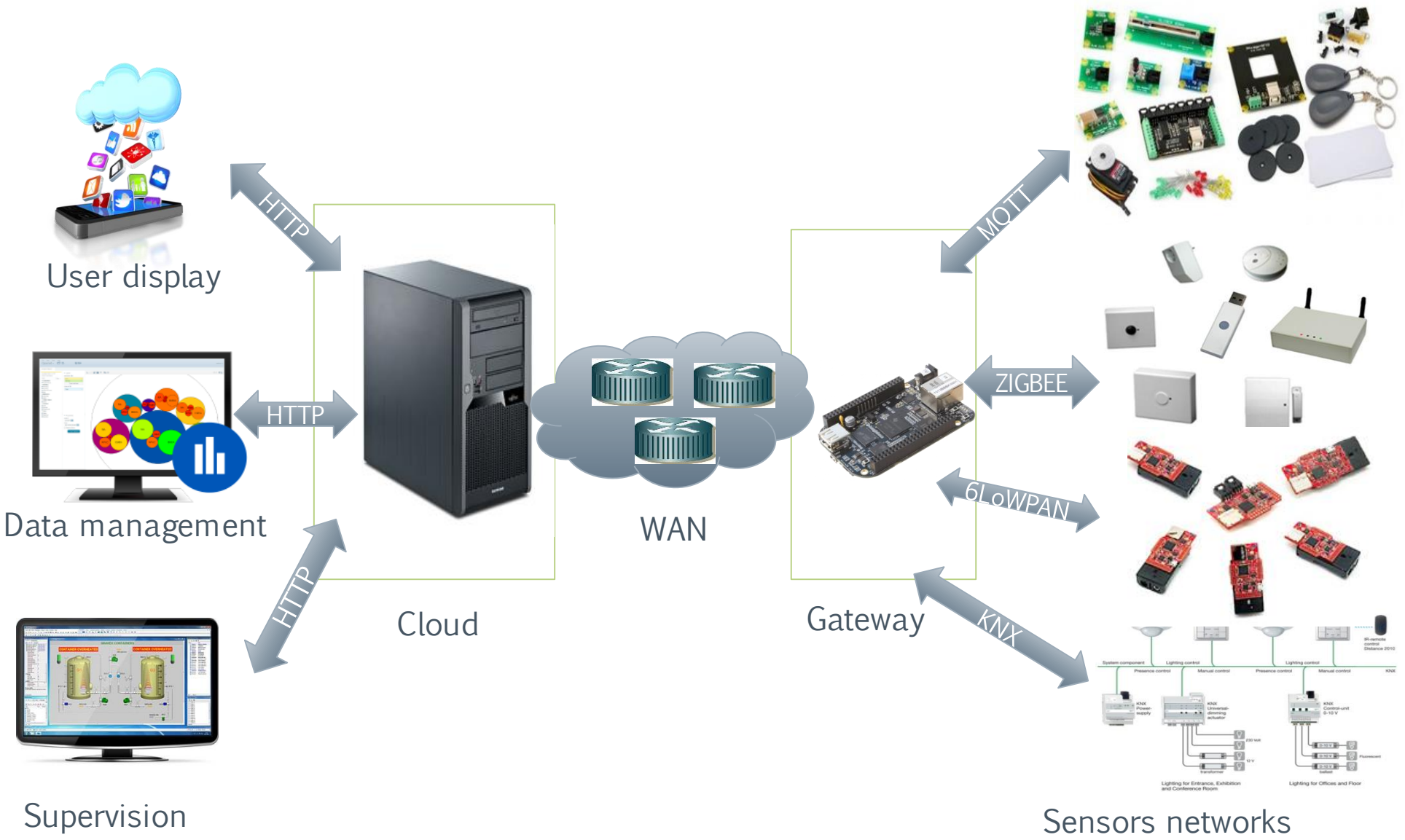


Use case: smart metering



- Obtain Meter Reading Data
- Install, Configure & Maintain the Smart Metering Information System
- Optimise the consumption
- Support Prepayment Functionality
- Manage Tariff Settings on the Smart Metering Information System
- Interact with Devices in the building
- Display Messages on consumption, advice
- ….

Common architecture



Market fragmentation

- The current marketplace is **extremely fragmented**
  - Increase the R&D cost in each specific domain.
  - **Silo model** is not an efficient way to communicate, it is a **barrier to further development**.
  - Many **vertical solutions** have been designed independently and separately for different applications, which impedes large-scale interoperable deployment.

IoT SDOs and Alliances Landscape

Home/Building

Manufacturing/ Industry Automation

Vehicular/ Transportation

Healthcare

Energy

Cities

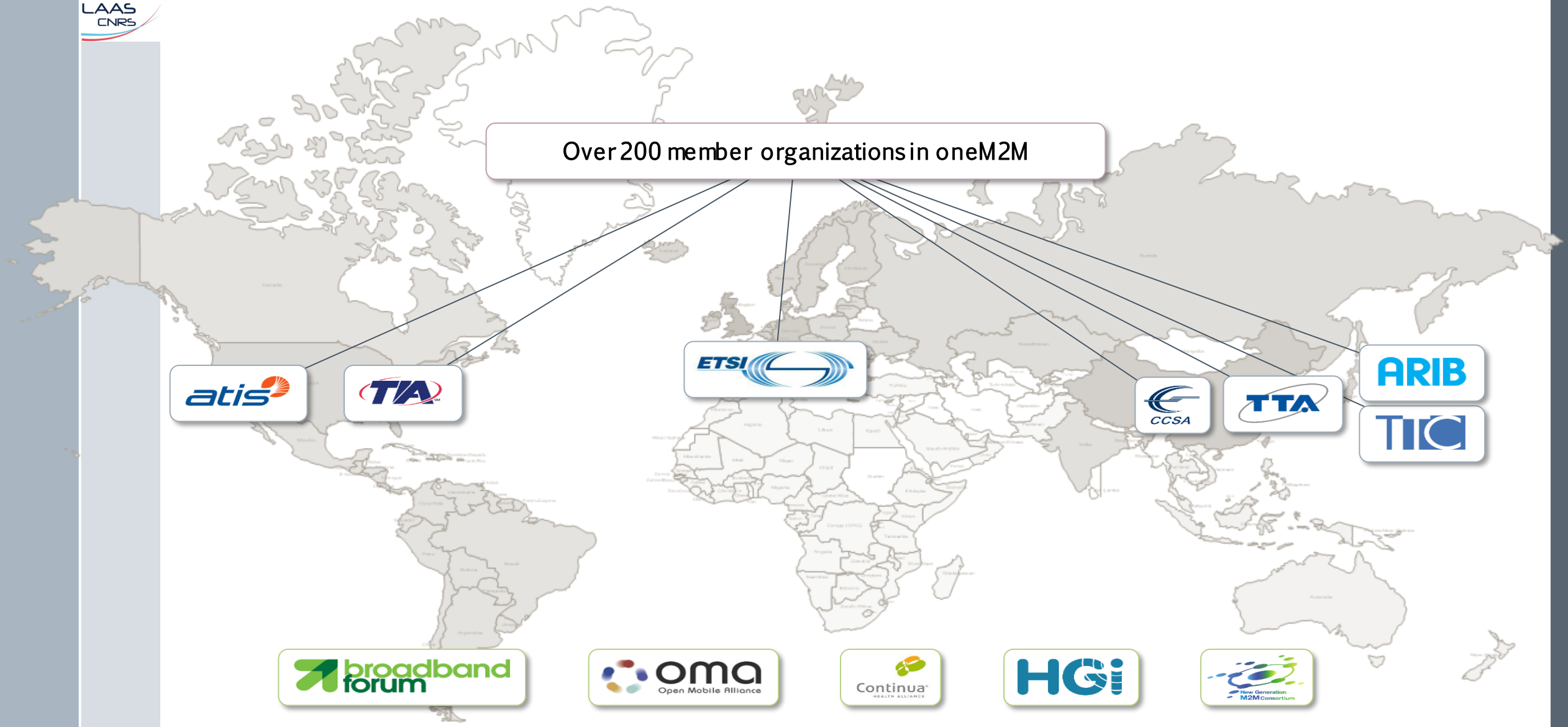
Wearables

Farming/ Agrifood

Horizontal/Telecommunication

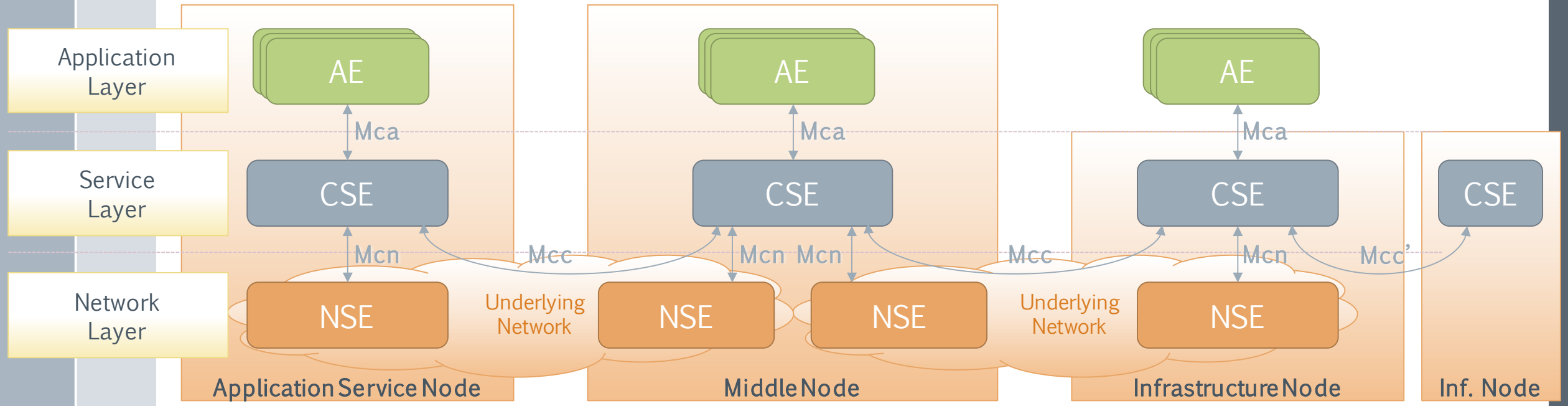
Source: AIOTI WG3 (IoT Standardisation)– Release 1.2

oneM2M: The Partnership Project



Standardized Architecture of OneM2M

Reference Point	One or more interfaces - Mca, Mcn, Mcc and Mcc' (between 2 service providers)
Common Services Entity	Provides the set of "service functions" that are common to the M2M environments
Application Entity	Provides application logic for the end-to-end M2M solutions
Network Services Entity	Provides services to the CSEs besides the pure data transport
Node	Logical equivalent of a physical (or possibly virtualized, especially on the server side) device



From oneM2M Service Layer Platform – Initial Release: Omar Elloumi/Nicolas Damour

Standardized OneM2M Service

Other
MN Node

ADN-AE

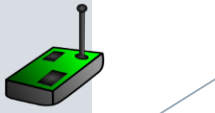
ADN Node

MN-AE

MN-CSE

SUB
REG
ASM
SEC
RF
LOC
Interworking
Proxy Entity
(IPE)
CMDH
DMR
GMG

MN Node



Non
oneM2M
Nodes

ASN-AE

ASN-CSE

SUB
REG
ASM
SEC
RF
LOC
DMR
GMG
CMDH

ASN
Node

ADN-AE

ADN Node

Other
IN Node

IN-AE

IN-CSE

Registration
(REG)
Subscription and
Notification (SUB)
Application and
Service Layer Mgmt
(ASM)
Security
(SEC)
Routing
Function
(RF)
Location
(LOC)
Data Mgmt and
Repository (DMR)
Group
Management
(GMG)
Communication Mgmt
& Delivery Handling
(CMDH)

IN
Node

REST Architecture Concepts

REST stands for (representational state transfer)

- **Resource** oriented
  - Stored on a server
- Access using an **URI**
  - http://www.example.com/wiki/rest
  - http://www.example.com/software/releases/latest.tar.gz
- **Representation** of resources
  - Used in exchange with client/user
  - Can be any representation format: XML, JSON, BSON, …
- **Link** to other resources
  - Dependencies, hierarchy is represented by link in resource representation

REST Architecture Properties

- **Addressability**
  - Each resource has unique URI
- **Statelessness**
  - Each requests contains all application states necessary to handle that request
- **Connectedness**
  - Resources are linked between each others
  - You can put the service in different states just by following links
- **Uniform interface:** based on HTTP operations
  - Retrieve a resource: HTTP **GET**
  - Create new resource: HTTP **POST**
  - Update a resource: HTTP **PUT**
  - Delete a resource: HTTP **DELETE**

oneM2M pros

- Standardized architecture
- Standardized Service
- Standardized API
- Standardized structuration of data

oneM2M provides interoperability

oneM2M cons

- oneM2M is a specification, not a software
- Lots of documents :
  - 24 specification documents,
  - 24 technical reports

How to use this standard ?

OM2M: Open platform for IoT eclipse.org/om2m

- **Standard benefits :**
  - **Compliant** to **SmartM2M** Standard (April 2014) and with **OneM2M** Standard (november 2015)
  - **Horizontal** service platform for IoT **interoperability**
  - **Restful** API with a **generic** set of service **capabilities**
  - Allow developing services **independently** of the underlying network
  - Facilitate **deployment** of **vertical** applications
  - Main features:

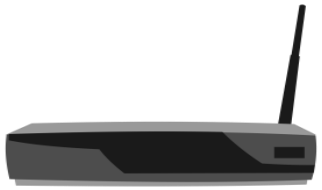
  Machine registration, application deployment, container management, resource discovery, access right authorization, subscription / notification, group management and non-blocking requests.

- **But also:**
  - **OSGi-based** architecture **extensible** via plugins
  - **Eclipse foundation project**
  - OM2M is an **open source project**
  - Member of Eclipse **IoT Working Group**.

Standardized OM2M resource

CSE | APPLICATION ENTITY | CONTAINER | CONTENT INSTANCE | REMOTE-CSE | SUBSCRIPTION

Smart Meter (ADN)



Gateway (MN-CSE)

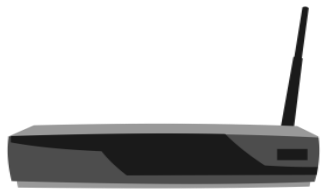
Server/Cloud (IN-CSE)

End user (DA)

OM2M resource tree example

CSE | APPLICATION ENTITY | CONTAINER | CONTENT INSTANCE | REMOTE-CSE | SUBSCRIPTION

MN-CSE-HOME

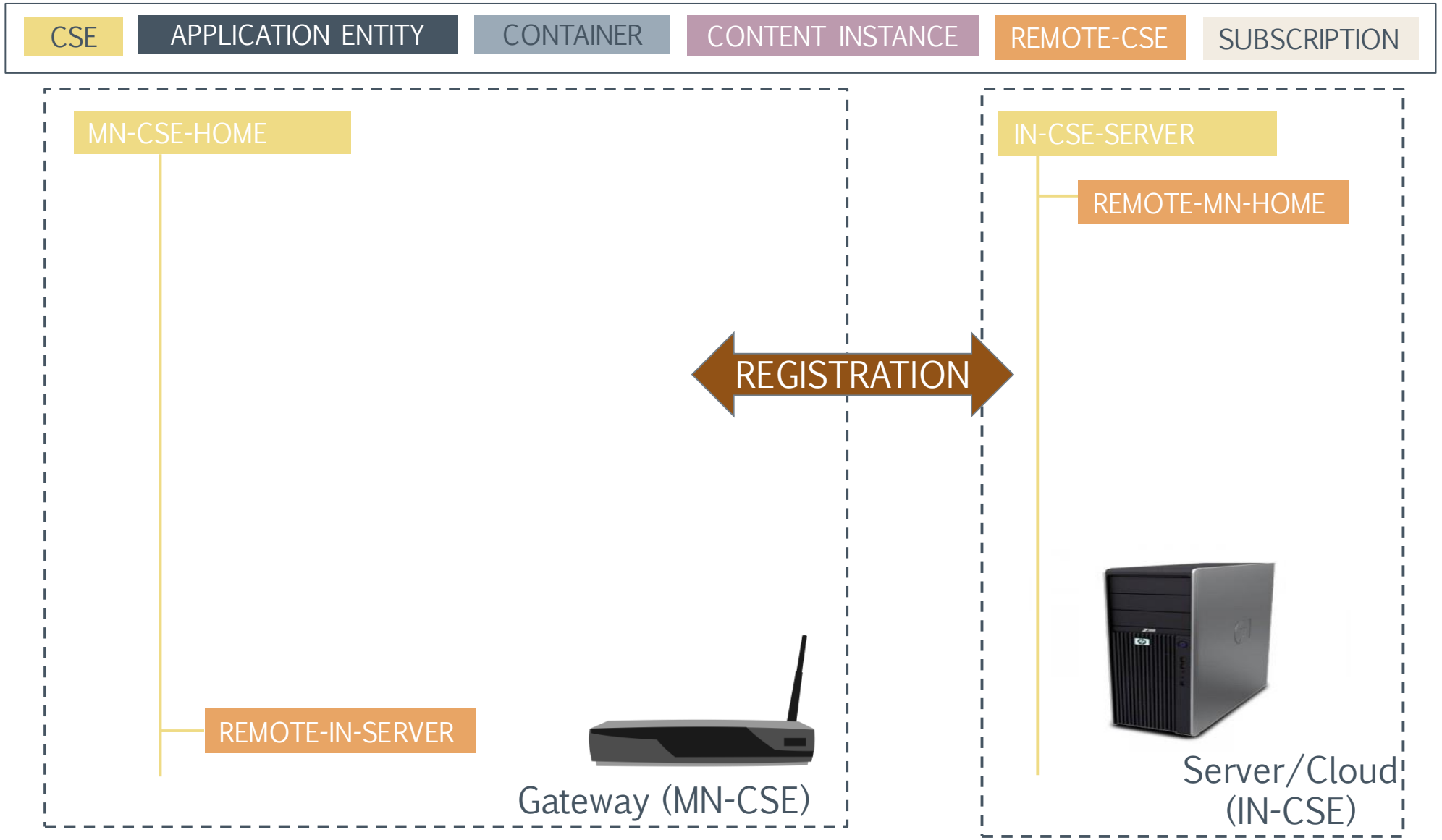


Gateway (MN-CSE)

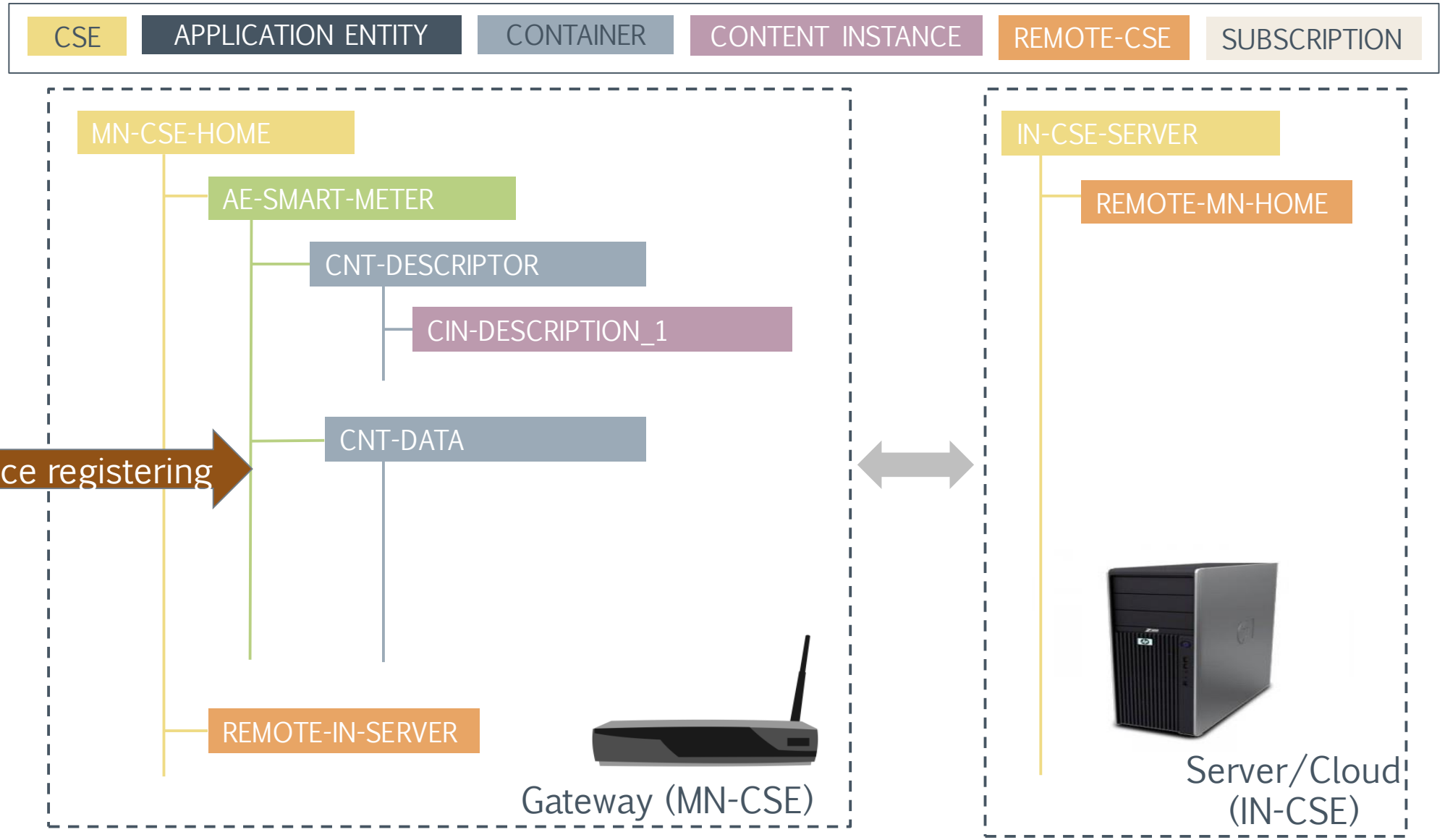
IN-CSE-SERVER

Server/Cloud
(IN-CSE)

OM2M resource tree example



OM2M resource tree example



OM2M resource tree example

Smart Meter (ADN)

New value

CSE | APPLICATION ENTITY | CONTAINER | CONTENT INSTANCE | REMOTE-CSE | SUBSCRIPTION

MN-CSE-HOME

AE-SMART-METER

CNT-DESCRIPTOR

CIN-DESCRIPTION_1

CNT-DATA

CIN-MEASUREMENT_1

REMOTE-IN-SERVER

Gateway (MN-CSE)

IN-CSE-SERVER

REMOTE-MN-HOME

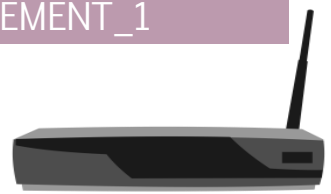
Server/Cloud (IN-CSE)

OM2M resource tree example

Smart Meter (ADN)

CSE | APPLICATION ENTITY | CONTAINER | CONTENT INSTANCE | REMOTE-CSE | SUBSCRIPTION

- MN-CSE-HOME
  - AE-SMART-METER
    - CNT-DESCRIPTOR
      - CIN-DESCRIPTION_1
    - CNT-DATA
      - CIN-MEASUREMENT_1
  - REMOTE-IN-SERVER



Gateway (MN-CSE)

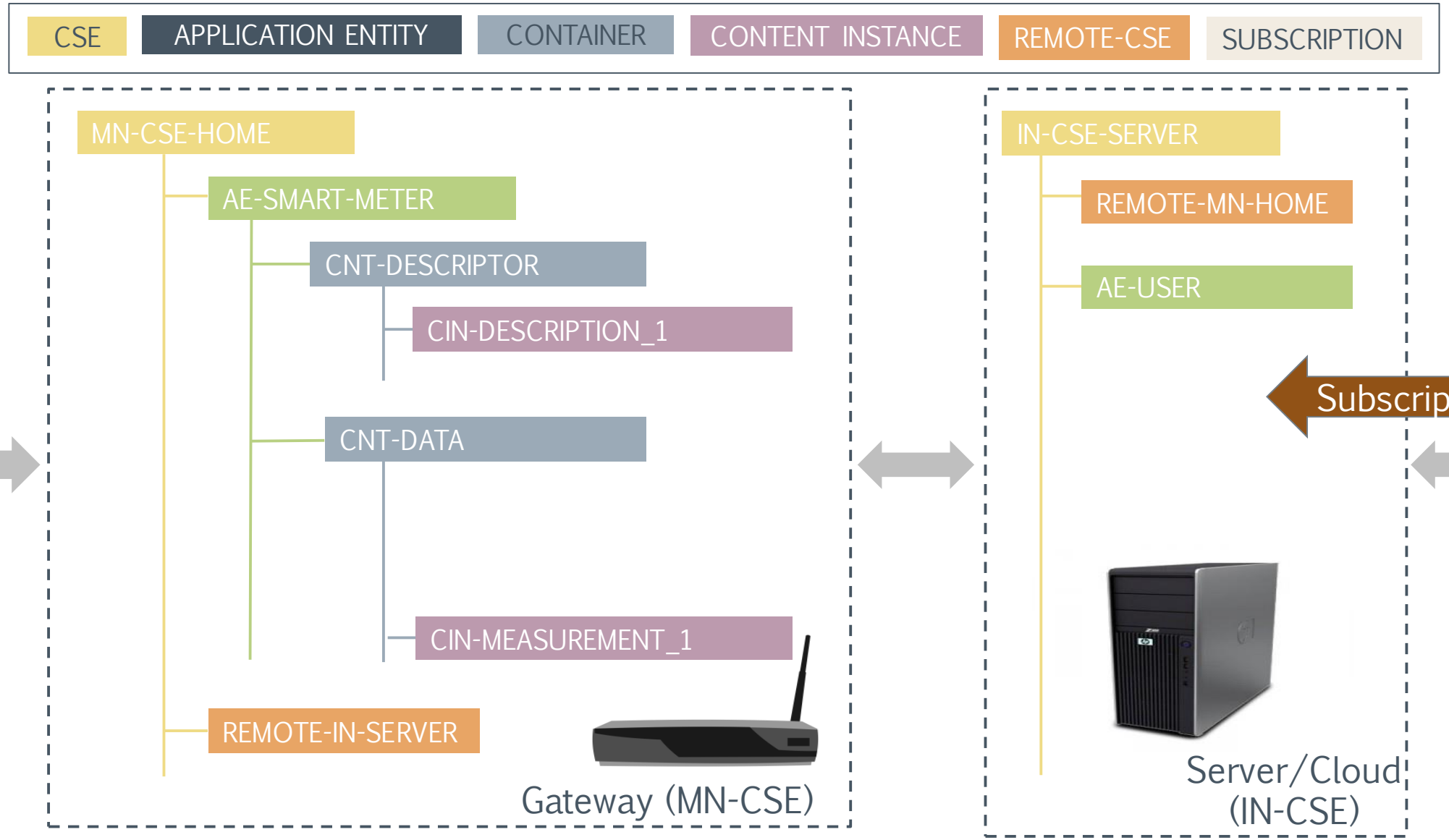
- IN-CSE-SERVER
  - REMOTE-MN-HOME
  - AE-USER

User registration

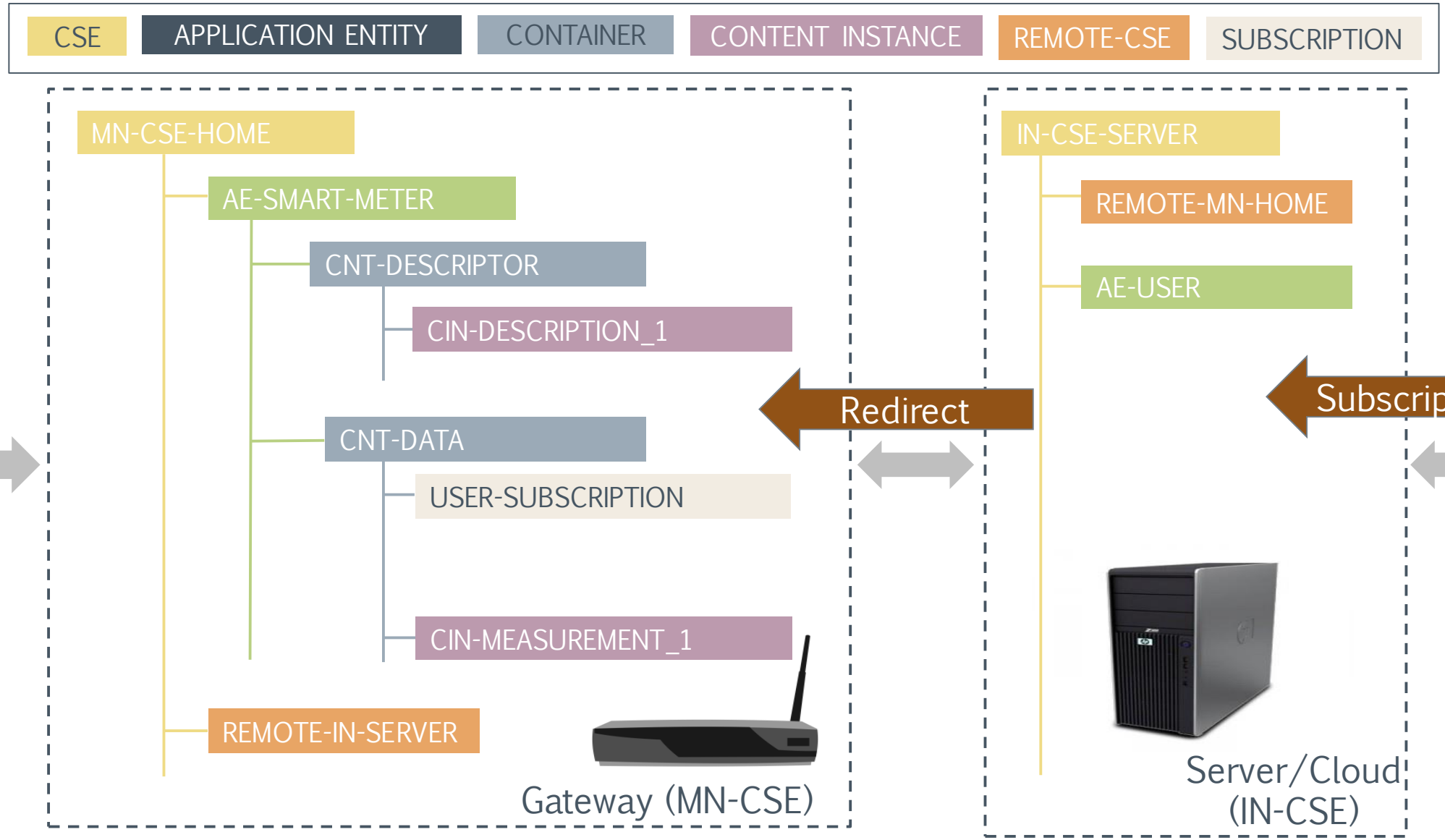
Server/Cloud (IN-CSE)

End user (DA)

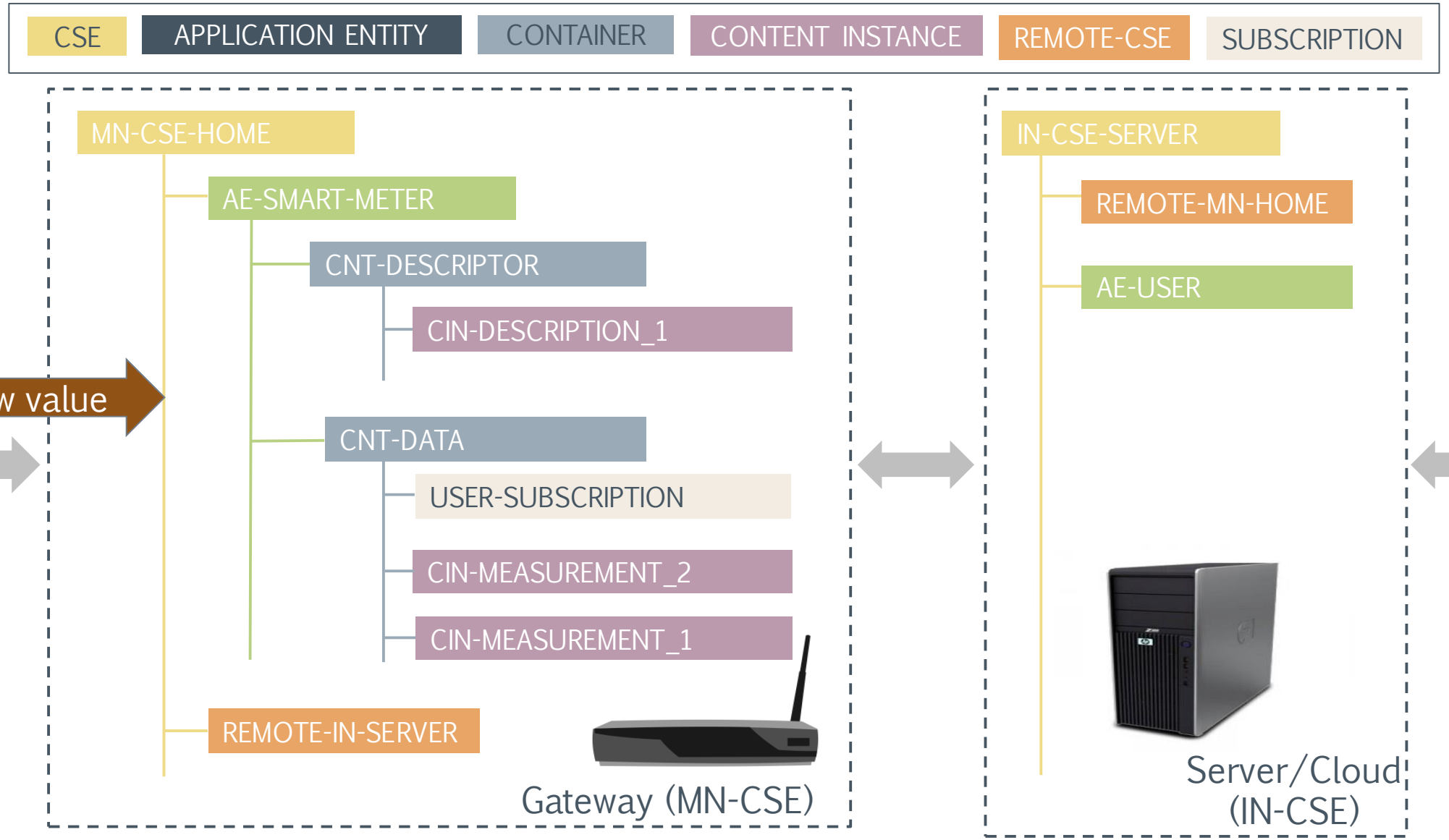
OM2M resource tree example



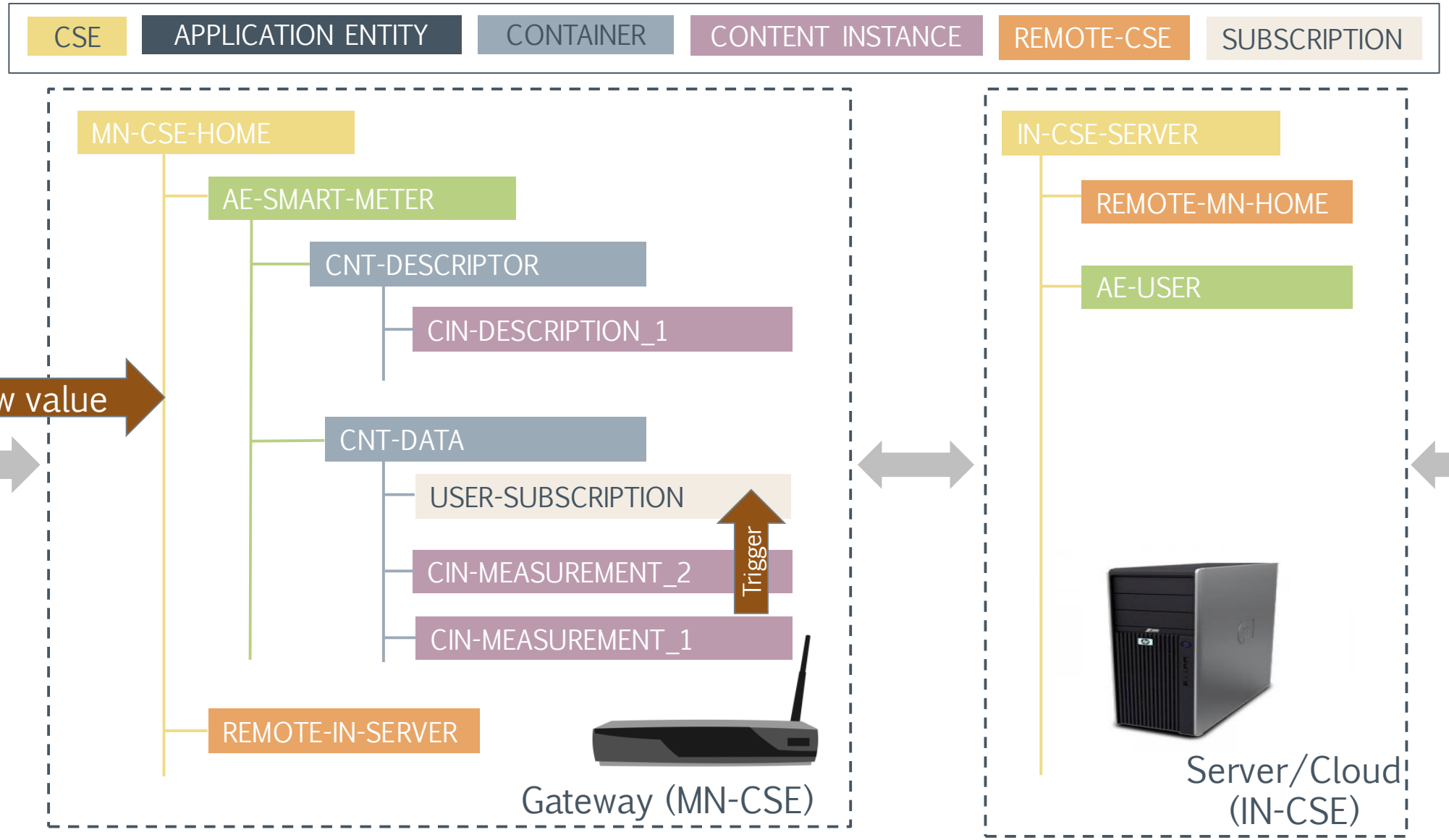
OM2M resource tree example



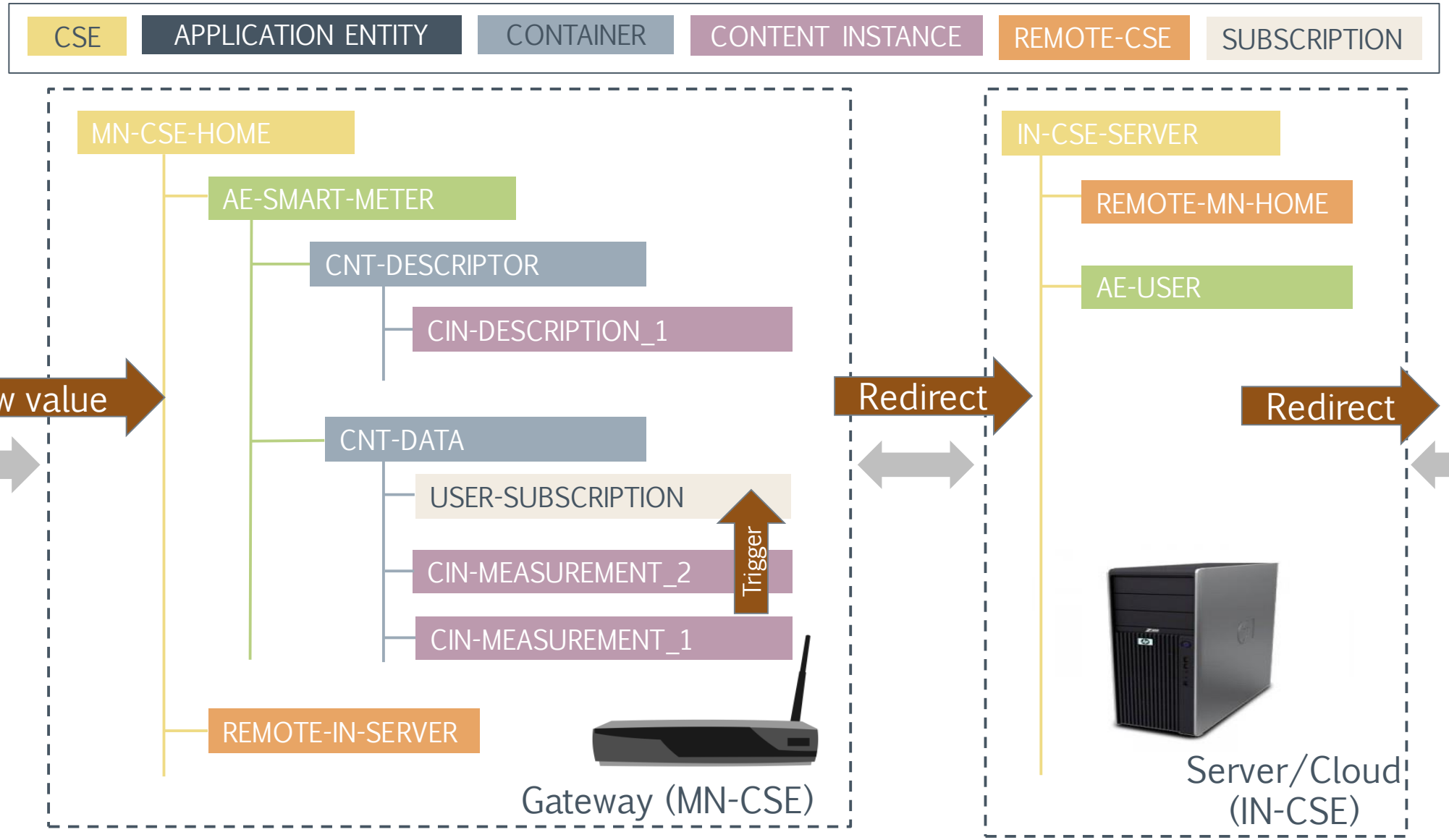
OM2M resource tree example



OM2M resource tree example



OM2M resource tree example



OM2M resource tree example

Smart Meter
(ADN)

CSE | APPLICATION ENTITY | CONTAINER | CONTENT INSTANCE | REMOTE-CSE | SUBSCRIPTION

- MN-CSE-HOME
  - AE-SMART-METER
    - CNT-DESCRIPTOR
      - CIN-DESCRIPTION_1
    - CNT-DATA
      - USER-SUBSCRIPTION
      - CIN-MEASUREMENT_2
      - CIN-MEASUREMENT_1
  - REMOTE-IN-SERVER

Gateway (MN-CSE)

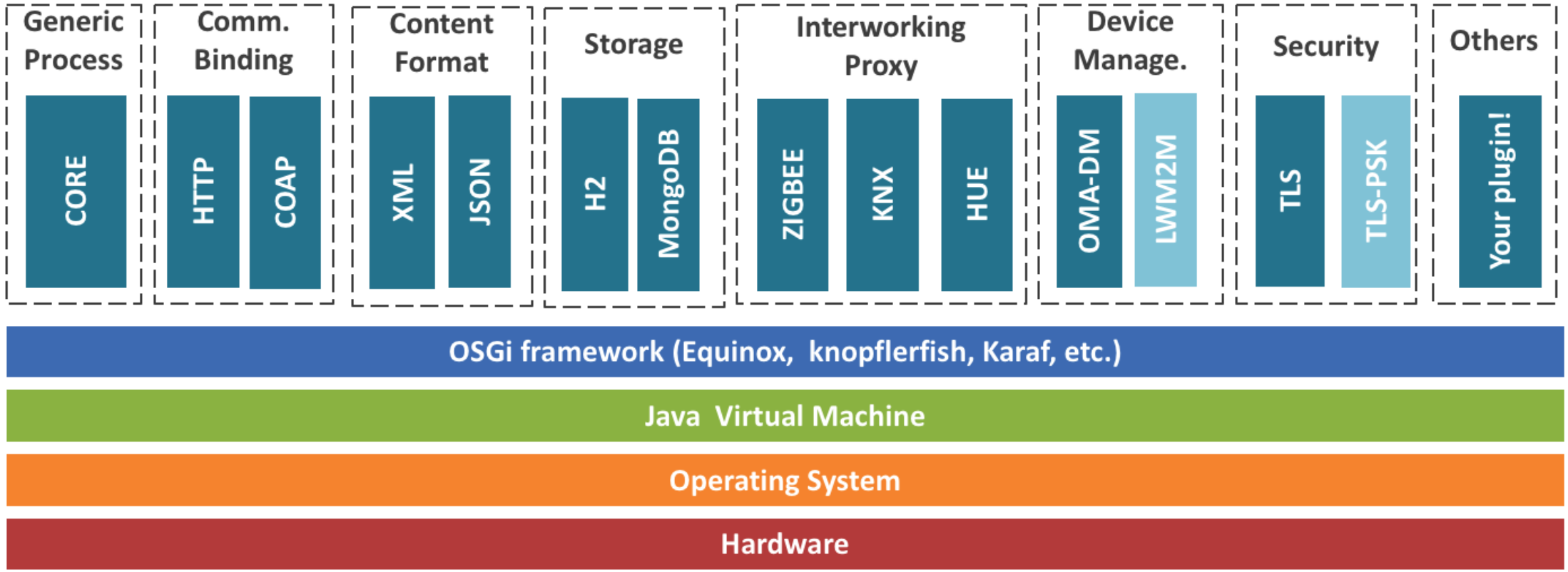
- IN-CSE-SERVER
  - REMOTE-MN-HOME
  - AE-USER

Server/Cloud
(IN-CSE)

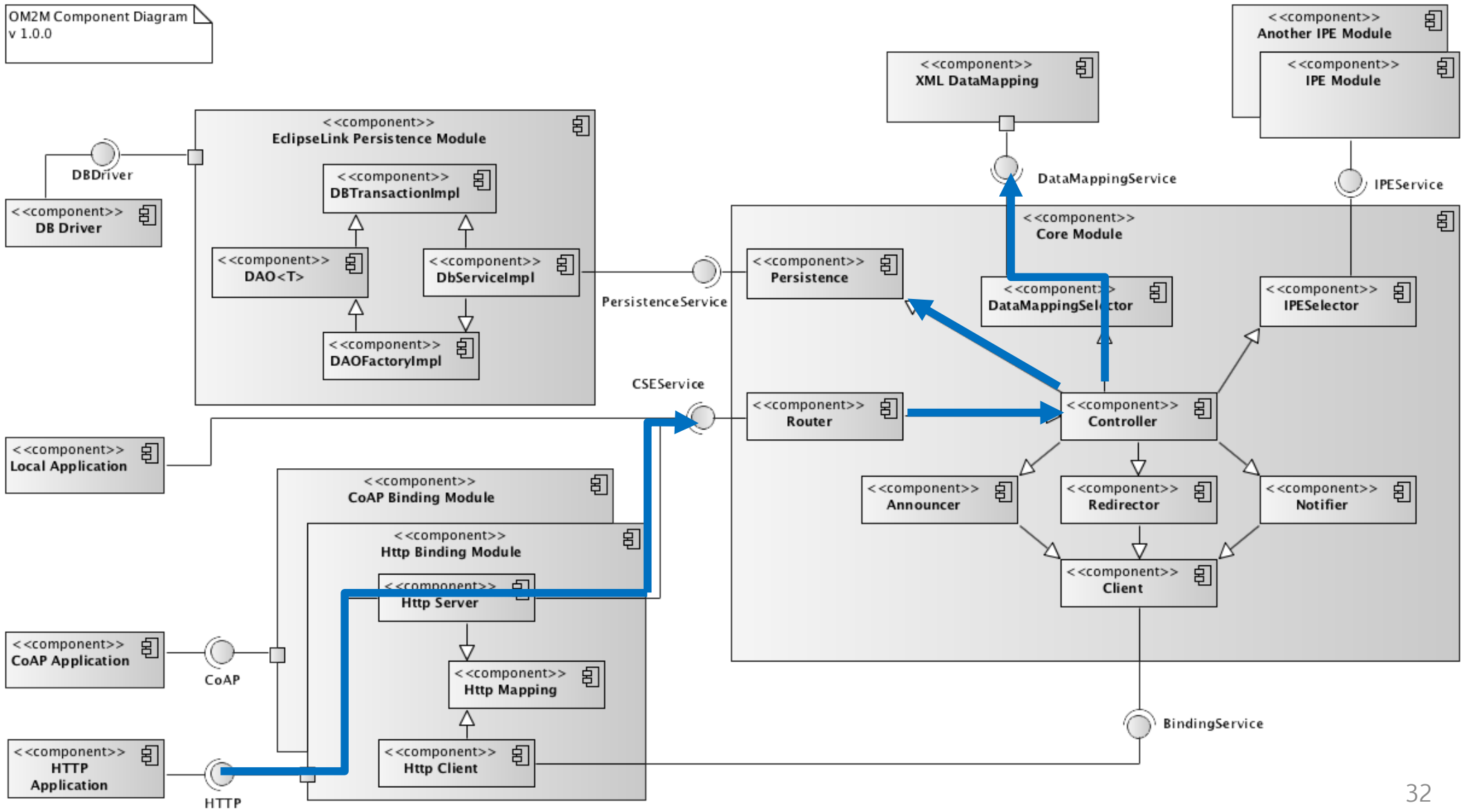
End user
(DA)

OM2M Building Blocks

- Java platform running on top of an **OSGi** runtime
  - Highly extensible via **plugins**
  - **Flexible OSGi container**: Equinox, Knopflerfish, or others.
  - **Flexible database** based on EclipseLink
- Build with **Maven** and **Tycho** for fast plugin developement



OM2M: Main components

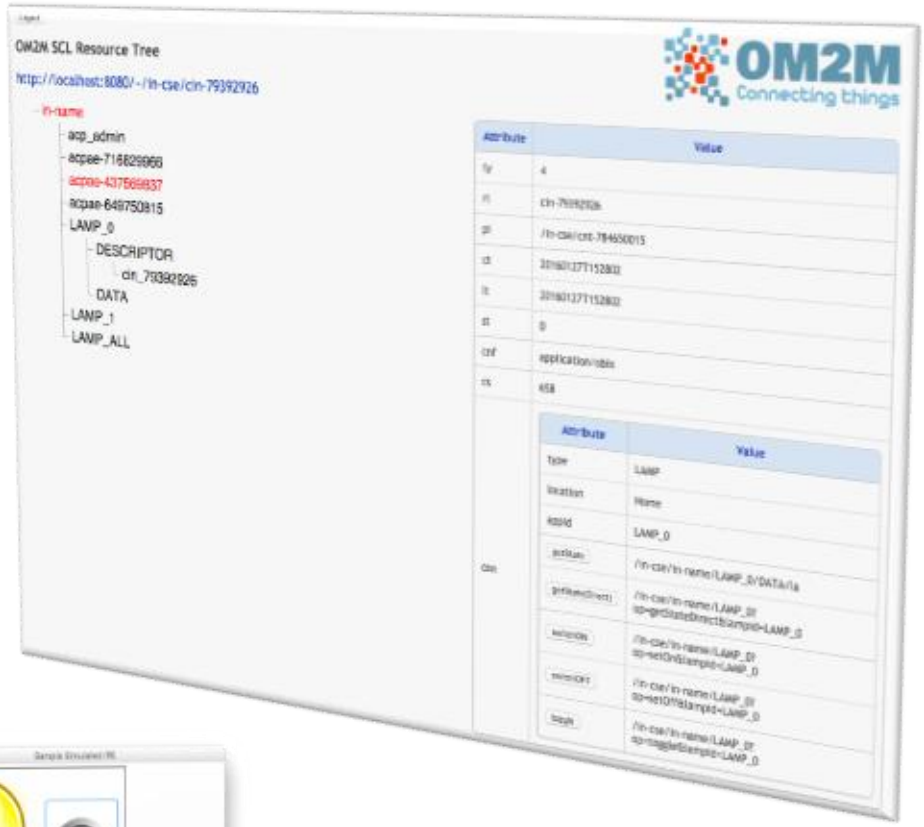


OM2M strengths

- **Open source** implementation of the standard
- Mechanisms to integrate **several** sensors **technologies**
- Creation of a community of users around the world
- Used in several universities to train students
  - At INSA: Innovative Smart System

OM2M: Web Resources

- tutorials
  - Clone & Build
  - Config & Start
  - Starting
  - Web Interface
  - REST API
  - Add your plugin
  - Interworking Proxy Entity



Simulated lamps

CSE | APPLICATION ENTITY | CONTAINER | CONTENT INSTANCE | REMOTE-CSE | SUBSCRIPTION

Lamp
(IPE)

MN-CSE-HOME

AE-LAMPS

DESCRIPTOR

CIN-DESCRIPTOR

DATA

CIN-VALUE_1

REMOTE-IN-SERVER

Gateway (MN-CSE)

IN-CSE-SERVER

REMOTE-MN-HOME

Server/Cloud
(IN-CSE)

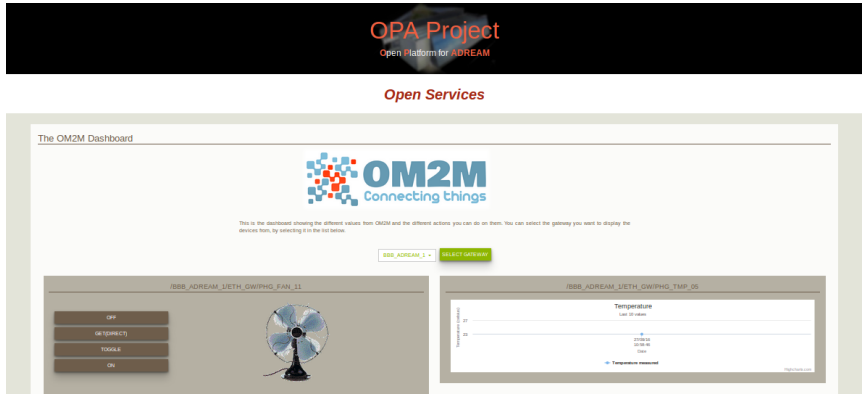
REST
client

Going beyond OM2M

- Graph-based energy management
- Autonomic computing
- Semantic data enrichment
- Open access to data and services
- …

Build easily any IoT application

Open service



- acpae-8049664
- acpae-558473692
- acpae-120601476
- PHL_LMP_01
- PHG_LMP_10
- PHL_LMP_02
- PHG_FAN_11
  - DATA
  - DESCRIPTOR
    - cin_390304589
  - LINE_INFO
- PHG_TMP_00
- PHG_HUM_01
- PHG_LUM_02
- PHG_WGH_03
- PHG_PRS_04

cs	588
con	

Attribute	Value
type	FAN
unit	
location	HOME
GET	/BBB_ADREAM_1/ETH_GW/PHG_FAN_11 /DATA/la
GET(Direct)	/BBB_ADREAM_1/ETH_GW /PHG_FAN_11?op=get
ON	/BBB_ADREAM_1/ETH_GW /PHG_FAN_11?op=true
OFF	/BBB_ADREAM_1/ETH_GW /PHG_FAN_11?op=false
TOGGLE	/BBB_ADREAM_1/ETH_GW /PHG_FAN_11?op=toggle

OM2M Connecting things

Conclusion

- Standards
  - can decrease the costs of development and maintenance
  - allow to have multiple suppliers
  - make it easier for users
- But difficulties to chose the good standard
- A **standard** like OneM2M allows to **hide the heterogeneity**
- Open source
  - increases the numbers of users and create communities
  - increases the visibility of your work
  - helps to **extend** the capabilities of the software
  - can create important feedback to standard
- But **open source contribution are different than proof of concept development**

Conclusion

- Adressed IoT challenges:
  - Data interoperability => semantic (Phd **N. Seydoux**)
  - Scalability => cloud for IoT service layer (PhD **Y. Banouar**)
  - Easy development => framework
  - Invisibility => standard and semantic
  - Network capabilities for IoT => LoRa, 5G ? (PhD **C. El Fehri,** post-doc **N. Accetura**)
  - Device management => Efficient management of system with LWM2M standard (Phd **G. Garzone**)
  - System management => use of checkpointing mechanism (Phd **F. Aïssaoui**)
  - Large domain of usage : smart city, smart grid, smart factory, etc => develop complex service based on multiple dynamic simple service (phd **G. Garzone**)

Thank you

OM2M
Connecting things