行動寬頻尖端技術跨校教學聯盟

M2M Service Architecture (2)

國立交通大學資訊工程系
Department of Computer Science
National Chiao Tung University
October 15, 2018

Outline

1. M2M Service Capabilities Framework
2. REST Architectural Style for M2M
3. Resource-Based M2M Communications

Resource-Based M2M Communications

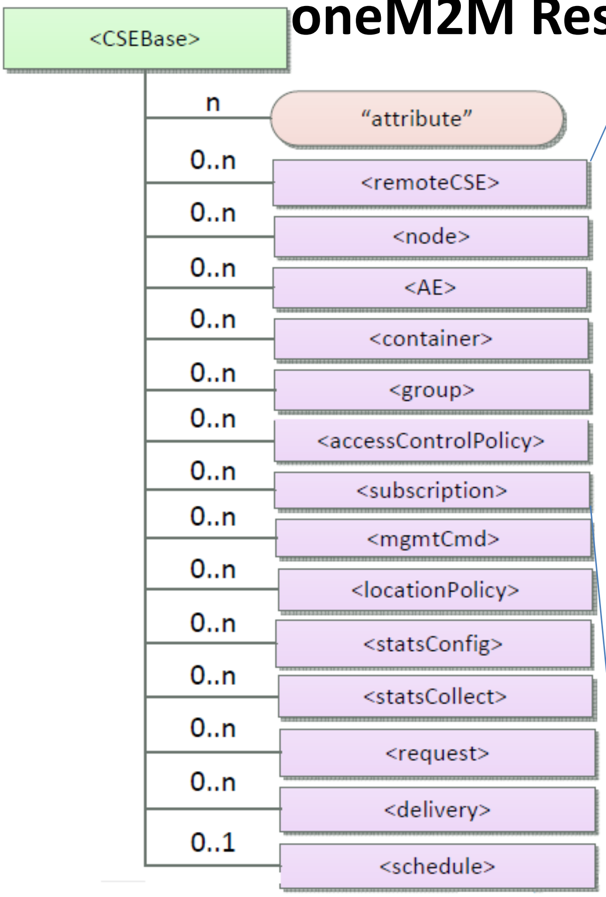
Resource-based Information Model

- All entities in the oneM2M System, such as AEs, CSEs, data, etc. are represented as resources.
- A resource structure is specified as a representation of such resources. Such resources are uniquely addressable via a Uniform Resource Identifier (URI).
- A given Resource is of one of the defined **Resource Types**.
- The Resource Type determines the semantics of the information in the Resource.

Properties of Resources

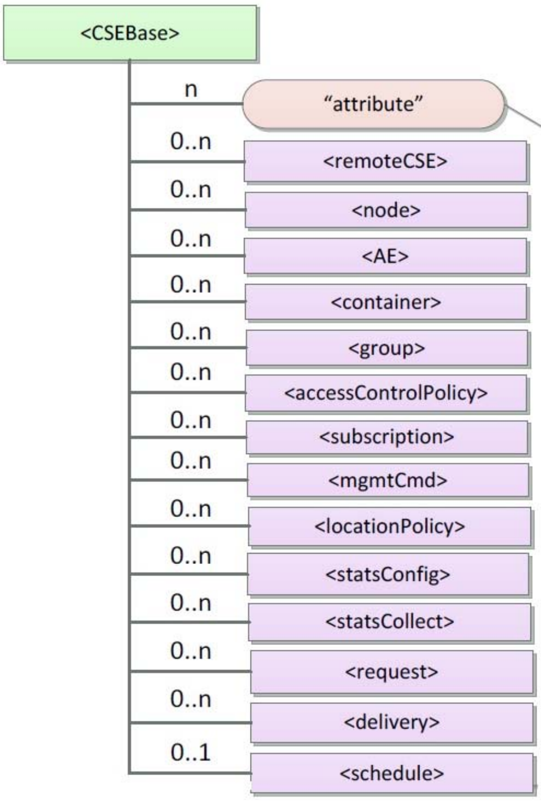
- Resources can be Created, Read, Updated or Deleted to manipulate the information.
- Resources are organized in a tree-like structure and connected by links.
- Addressing all resources and associated attributes through unique URI.
  - Structured URI based on the chain of parent-child relationship
  - Flat URI made of a unique identifier addressable via the base root

oneM2M Resource Structure (1)



Resource Type	Description
CSEBase	The structural root for all the resources that are residing on a CSE
remoteCSE	Represents a remote CSE registered on the registrar CSE identified by the CSEBase resource
node	Represents specific Node information
AE	Stores information about the AE. It is created as a result of successful registration of an AE with the registrar CSE
container	Shares data instances among entities. Used as a mediator that takes care of buffering the data to exchange "data" between AEs and/or CSEs.
group	Stores information about resources of the same type that need to be addressed as a Group
accessControlPolicy	It controls "who" is allowed to do "what" and the context in which it can be used for accessing resources
subscription	represents the subscription information related to a resource.

oneM2M Resource Structure (2)



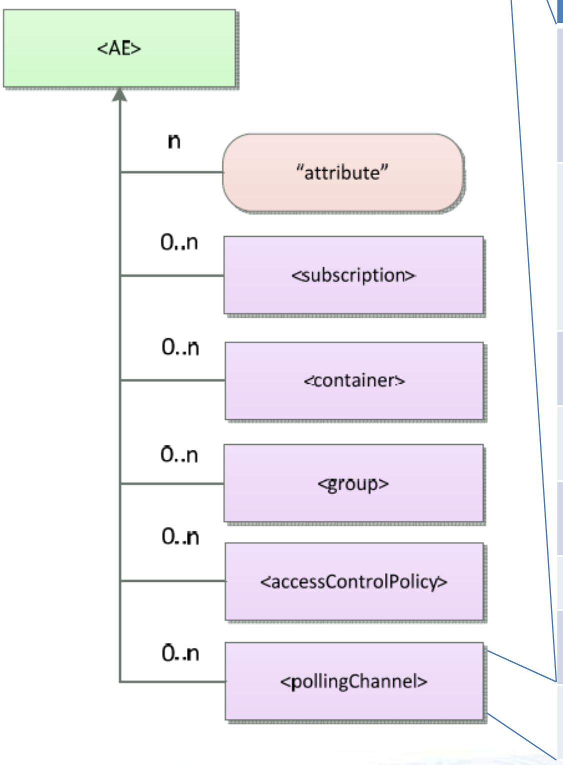
Common Attributes:

- resourceType
- resourceID
- parentID
- expirationTime
- accessControlPolicyIDs
- labels
- creationTime
- lastModifiedTime
- stateTag
- announceTo
- announcedAttribute

CSEBase Specific Attributes:

- cseType
- pointOfAccess
- CSEBase
- CSE-ID
- M2M-Ext-ID
- TriggerRicipient-ID
- requestReachability
- nodeLink

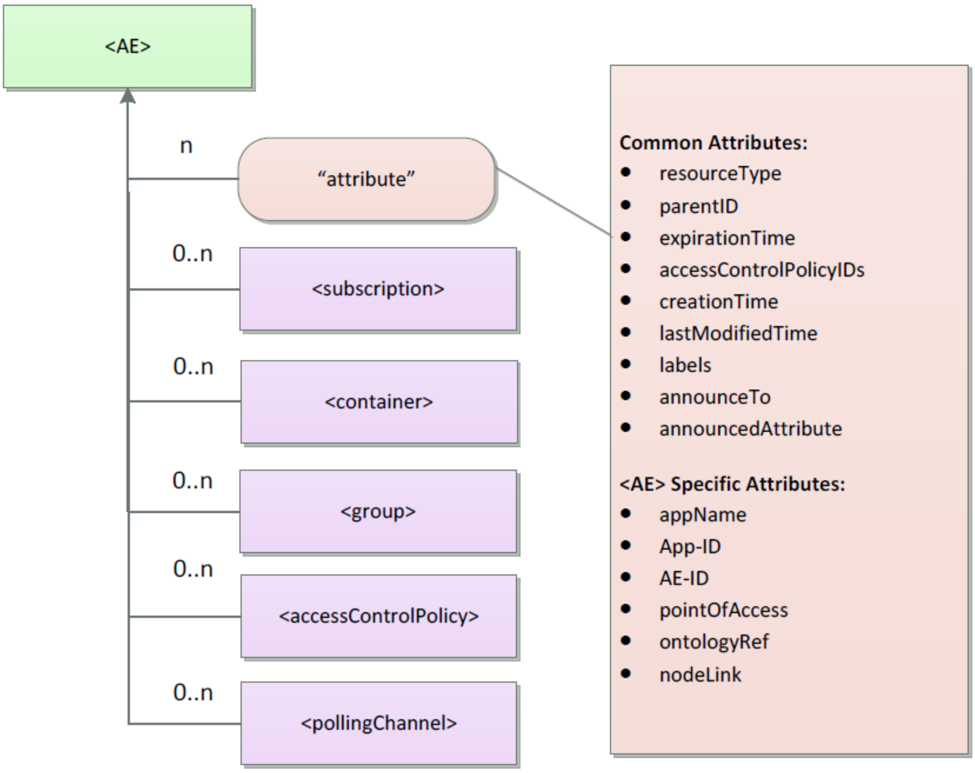
oneM2M Resource Structure (3)



Resource Type (Cont.)	Description
mgmtCmd	Management Command resource represents a method to execute management procedures required by existing management protocols.
locationPolicy	Includes information to obtain and manage geographical location. It is only referenced within a container, the contentInstances of the container provide location information
statsconfig	Stores configuration of statistics for applications
StatsCollect	Defines triggers for the IN-CSE to collect statistics for applications
request	Expresses/access context of an issued Request
delivery	Forwards requests from CSE to CSE
schedule	Contains scheduling information for delivery of messages
pollingChannel	Represent a channel that can be used for a request-unreachable entity

Details in "Table 9.6.1.1-1 Resource Types" of oneM2M-TS0001

oneM2M Resource Structure (4)



<AE>

n "attribute"

0..n <subscription>

0..n <container>

0..n <group>

0..n <accessControlPolicy>

0..n <pollingChannel>

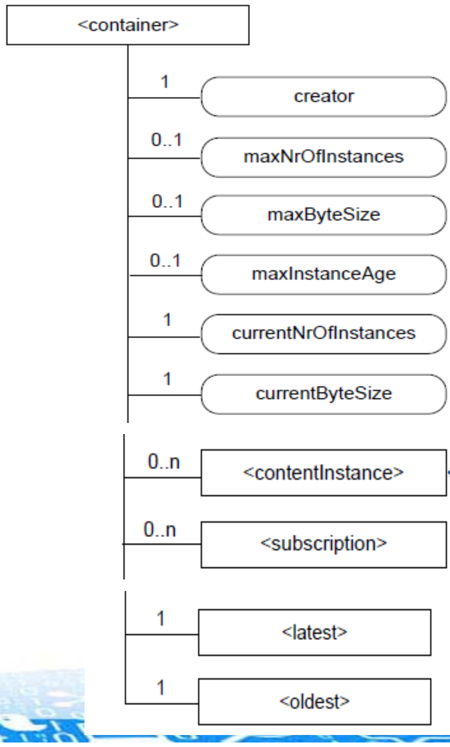
Common Attributes:

- resourceType
- parentID
- expirationTime
- accessControlPolicyIDs
- creationTime
- lastModifiedTime
- labels
- announceTo
- announcedAttribute

<AE> Specific Attributes:

- appName
- App-ID
- AE-ID
- pointOfAccess
- ontologyRef
- nodeLink

oneM2M Resource Structure (5)



<container>

1 creator

0..1 maxNrOfInstances

0..1 maxByteSize

0..1 maxInstanceAge

1 currentNrOfInstances

1 currentByteSize

0..n <contentInstance>

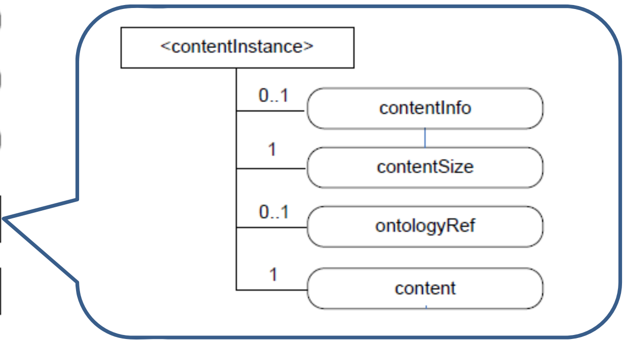
0..n <subscription>

1 <latest>

1 <oldest>

Container & ContentInstance

where the actual data is stored!



<contentInstance>

0..1 contentInfo

1 contentSize

0..1 ontologyRef

1 content

Resource Categories

oneM2M identifies three categories of resources:

- **Normal resources**
  - include the complete set of representations of data which constitutes the base of the information to be managed.
- **Virtual resources**
  - used to trigger processing and/or retrieve results, but they do not have a permanent representation in a CSE.
- **Announced resources**
  - a resource at a remote CSE that is linked to the original resource that has been announced, and it maintains some of the characteristics of the original resource.

Virtual Resource or Attribute

- A virtual resource or a virtual attribute is used to trigger processing and/or retrieve results, but they do not have a permanent representation in a CSE, for example:
  - **FanOutPoint** virtual resource is used for addressing bulk operations to all the resources that belong to a group.
  - **Latest/Oldest** virtual resources are pointers to the actual latest and oldest Content Instance in a Container.

Resource Addressing

- The address of a resource is a string that uniquely identifies the targeted resource within the scope of a request.
- Requests can have 3 different scopes:
  - **CSE-relative**, the request is originated within the same CSE as the targeted resource.
  - **SP-relative**, the request is originated within the same M2M Service Provider domain as the targeted resource, but at different CSE.
  - **Absolute**, the request is originated within an M2M Service Provider domain, but the target resource resides in a different M2M Service Provider domain.

Resource Addressing (cont'd)

Two different methods for addressing a resource:

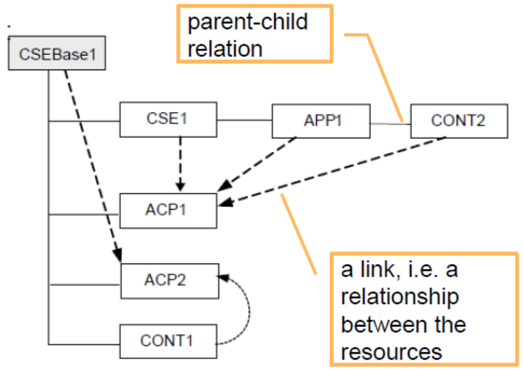
- Hierarchical URI (structured)
- Non-Hierarchical URI (non-structured)

Each addressing method can have three variants depending on the scope of the request:

	CSE-relative	SP-relative	Absolute
Hierarchical	streetX/houseY/roomZ/temp123	**/MN-CSE-02/**streetX/houseY/roomZ/temp123	**//m2m.com/MN-CSE-02/**streetX/houseY/roomZ/temp123
Non-Hierarchical	temp123	**/MN-CSE-02/**temp123	**//m2m.com//MN-CSE-02/**tempe123

Relationship between Resources

- Two types of relationships: parent-child and link



parent-child relation

a link, i.e. a relationship between the resources

Source: oneM2M TS-0001

Origin Resource type	Destination Resource type	Relationship type
Any (e.g. AE, container)	Access Control Policy	Link (access Control Policy ID)
CSEBase or remoteCSE	node	Link (nodeLink)
node	CSEBase or remoteCSE	Link (hosted CSE Link or CSEBase)
a child resource of any resource type	a parent resource of any resource type	Link (parent-ID)
a parent resource of any resource type	a child resource of any resource type	Parent-Child
mgmtObj	mgmtObj	Link (mgmtLink)

Normal (Virtual) Resource Types (1)

1. Access Control Policy
2. Content Instance
3. AE
4. Container
5. CSE Base
6. Delivery
7. Event Config
8. Exec Instance
9. Group
10. Location Policy
11. Latest (virtual)
12. Fan Out Point (virtual)
13. mgmtCmd
14. mgmtObj
15. M2m Service Subscription Profile
16. Node
17. Oldest (virtual)
18. Polling Channel
19. Polling Channel URI (virtual)
20. Remote CSE

Normal Resource Types (2)

21. Request
22. Schedule
23. Service Subscribed Node
24. *Service Subscribed App Rule*
25. Stats Collect
26. Stats Config
27. Subscription

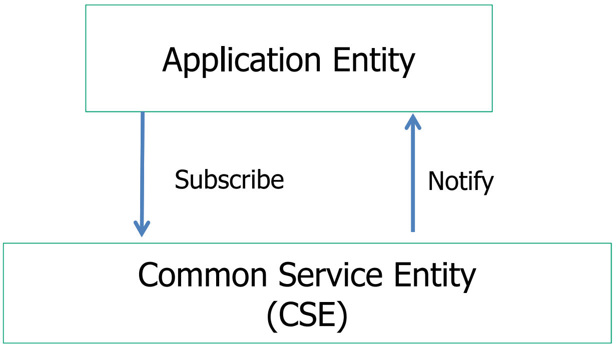
Resource Type Specializations

Used by Communication Management and Delivery Handling or *Device Management CSFs*

1. *ActiveCmdhPolicy*
2. *Area Network Device Info*
3. *Area Network Info*
4. *Battery*
5. *cmdhBuffer*
6. *cmdhDefaults*
7. *cmdhEcDefParamValues*
8. *cmdhDefEcValue*
9. *cmdhLimits*
10. *cmdhNetworkAccessRules*
11. *cmdhNwAccessRule*
12. *cmdhPolicy*
13. *Device Capability*
14. *Device info*
15. *Event Log*
16. *Firmware*
17. *Memory*
18. *Reboot*
19. *Software*

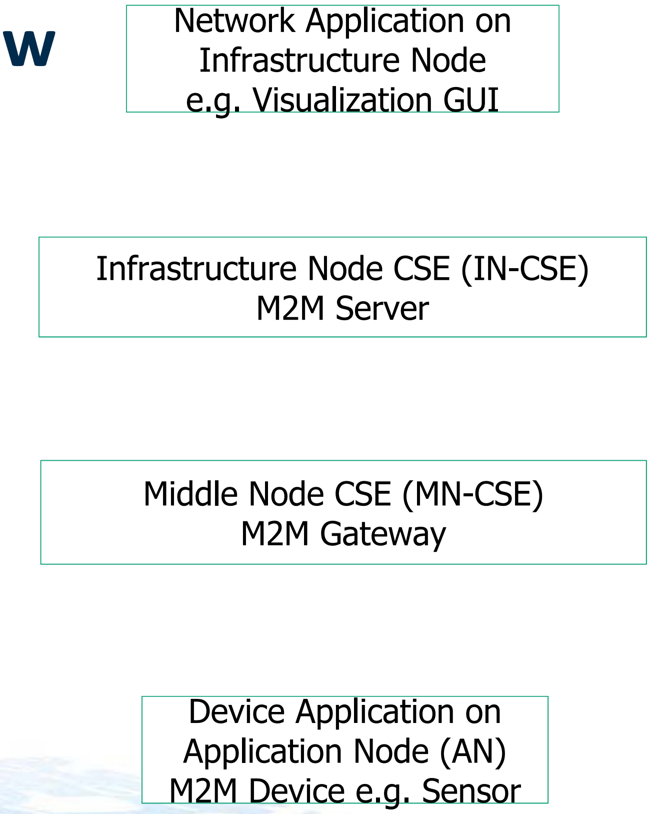
oneM2M Communication Concepts
Subscribe & Notify

- Data is often provided in irregular intervals.
- To alleviate the need for constant polling, a subscribe/notify mechanism is used.
- Notifications can be twofold:
  - Synchronous (long polling)
  - Asynchronous (server capable client)

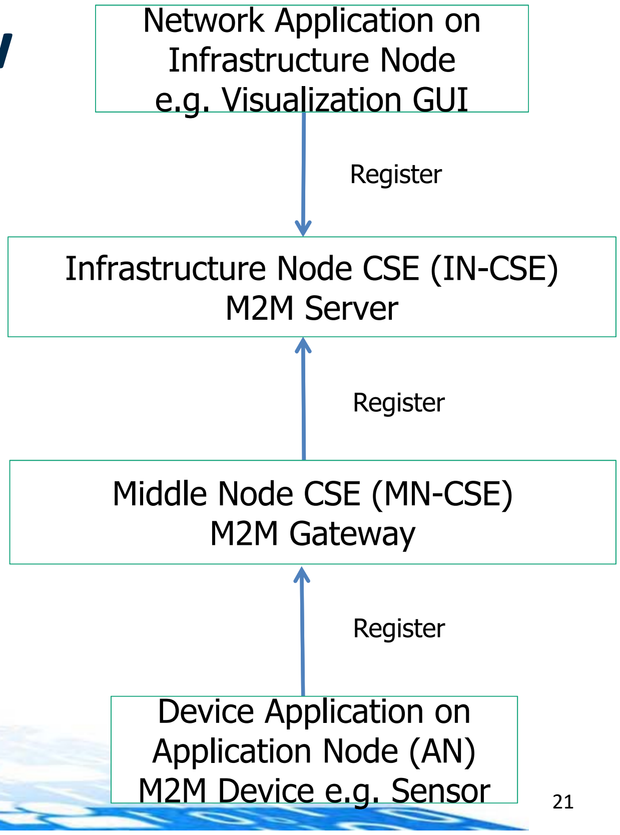


M2M End to End Communication
High Level Overview

- For a simple end to end example consider the following scenario:
- A Device application, for example a SmartMetering Sensor, is connected to a local M2M gateway.
- A Network Application, e.g. a SmartMeter GUI, is connected to a IN-CSE for visualizing sensor data.


Network Application on Infrastructure Node e.g. Visualization GUI
Infrastructure Node CSE (IN-CSE) M2M Server
Middle Node CSE (MN-CSE) M2M Gateway
Device Application on Application Node (AN) M2M Device e.g. Sensor

M2M End to End Communication High Level Overview



Network Application on Infrastructure Node e.g. Visualization GUI

Register

Infrastructure Node CSE (IN-CSE) M2M Server

Register

Middle Node CSE (MN-CSE) M2M Gateway

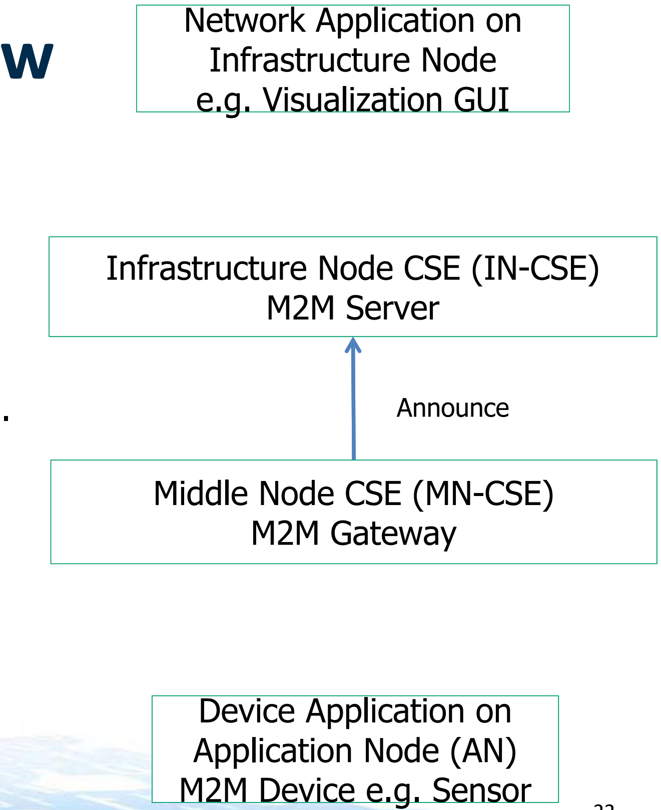
Register

Device Application on Application Node (AN) M2M Device e.g. Sensor

Registration Phase

- NA and DA register at their respective local CSEs.
- The MN-CSE registers at the IN-CSE

M2M End to End Communication High Level Overview



Network Application on Infrastructure Node e.g. Visualization GUI

Infrastructure Node CSE (IN-CSE) M2M Server

Announce

Middle Node CSE (MN-CSE) M2M Gateway

Device Application on Application Node (AN) M2M Device e.g. Sensor

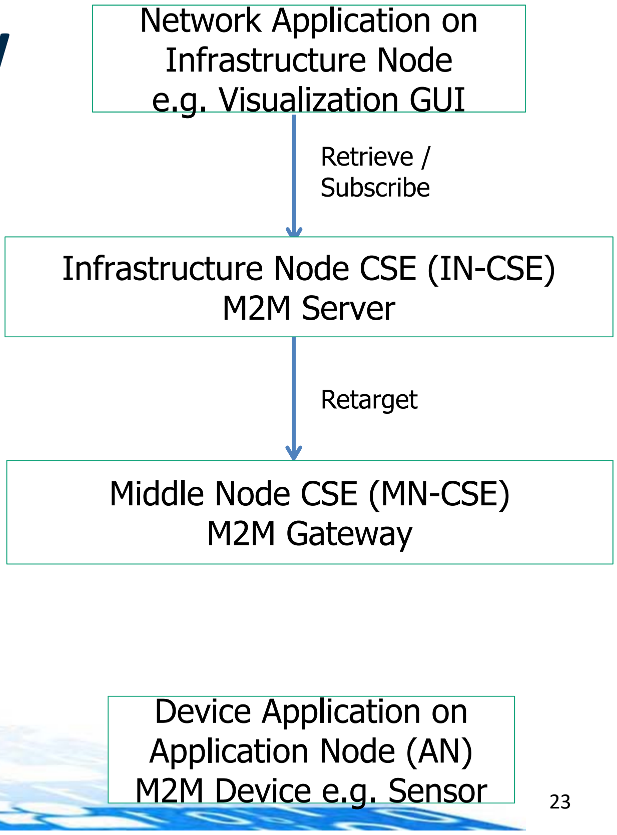
Announcement

- MN-CSE announces local applications towards the IN-CSE.

M2M End to End Communication High Level Overview

- The NA retrieves information on existing applications -> receives information about the sensor DA
- If neccessary, the request is retargeted by the IN-CSE to the MN-CSE
- Subsequently, the NA subscribes to the DA's data in order to be notified when new data arrives.

Source: Tutorial from FOKUS



Network Application on Infrastructure Node e.g. Visualization GUI

Retrieve / Subscribe

Infrastructure Node CSE (IN-CSE) M2M Server

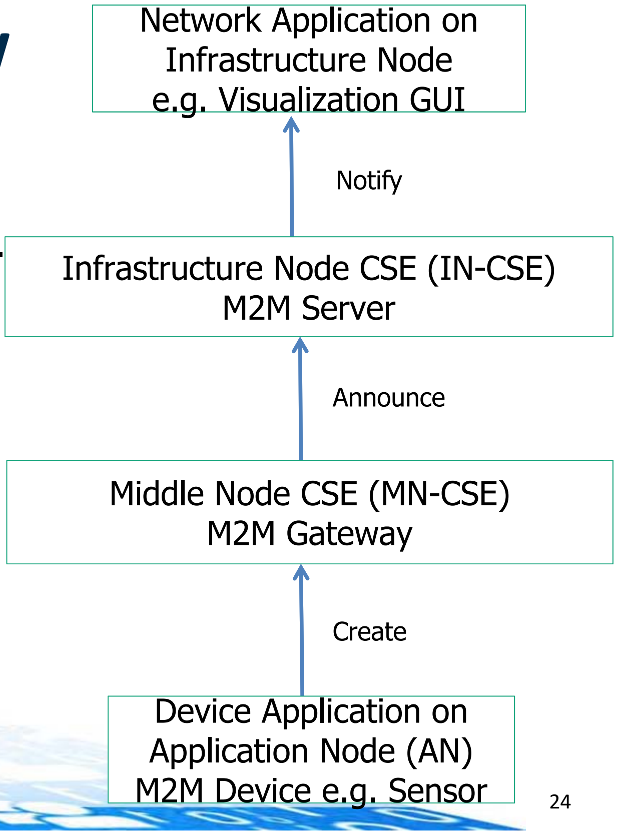
Retarget

Middle Node CSE (MN-CSE) M2M Gateway

Device Application on Application Node (AN) M2M Device e.g. Sensor

M2M End to End Communication High Level Overview

- Periodically, the sensor DA pushes metering data to the local gateway.
- This is done by creating a „contentInstance"
- The incoming data is announced towards the IN-CSE.
- The NA is notified about the incoming data.



Network Application on Infrastructure Node e.g. Visualization GUI

Notify

Infrastructure Node CSE (IN-CSE) M2M Server

Announce

Middle Node CSE (MN-CSE) M2M Gateway

Create

Device Application on Application Node (AN) M2M Device e.g. Sensor

Filtering

- Often, not all information in large data sets is required.
  - Inefficient to always transmit all data
- Request issuer can filter results by applying filter criteria
- Filter criteria can be applied in subscriptions and retrievals
  - Uses query string for retrievals
  - Embedded into subscription

Filtering through FilterCriteria

- Filter criteria can optionally specified to filter the notification result
  - Represented as an attribute of the subscription
  - Or as request parameters to RETRIEVE requests
    - E.g. URL parameters
- Filter criteria are applied to elements of the result sets
  - If multiple filter criteria are specified, only one needs to match (OR)
- If no filter criteria match, no notification is sent
- Extened set of filter criteria available for content instances

General Communication Flow

- The general flow is based on the use of Request and Response messages.
- The message applies to communications such as:
  - between an AE and a CSE (Mca reference point); and
  - among CSEs (Mcc reference point).
- Such communications can be initiated either by the AEs or by the CSEs depending upon the operation in the Request message.

Request

The Request from an Originator to a Receiver includes the following parameters:

- **ri:** Request Identifier.
- **to:** URI of the target resource for the operation.
- **fr:** Identifier representing the Originator.
- **cn:** resource content to be transferred.
- **role:** optional, required when role based access control is applied (associated text and procedure TBD).
- **op:** operation to be executed: **Create (C), Retrieve (R), Update (U), Delete (D), Notify (N)**

Types of Requests

Four Classic REST Operations:

1. CREATE: Create child resources (SUBSCRIBE is a CREATE).
2. RETRIEVE: Read the content of the resource.
3. UPDATE: Write the content of the resource.
4. DELETE: Delete the resource.

Two Extended Operations:

1.NOTIFY: indicate a change of a resource for a subscription. Variant of RETRIEVE/UPDATE.

2.EXECUTE: for executing a management command/task which is represented by a resource, corresponding to an UPDATE method without payload data.

Response

The Response from a Receiver to the Originator includes the following parameters:

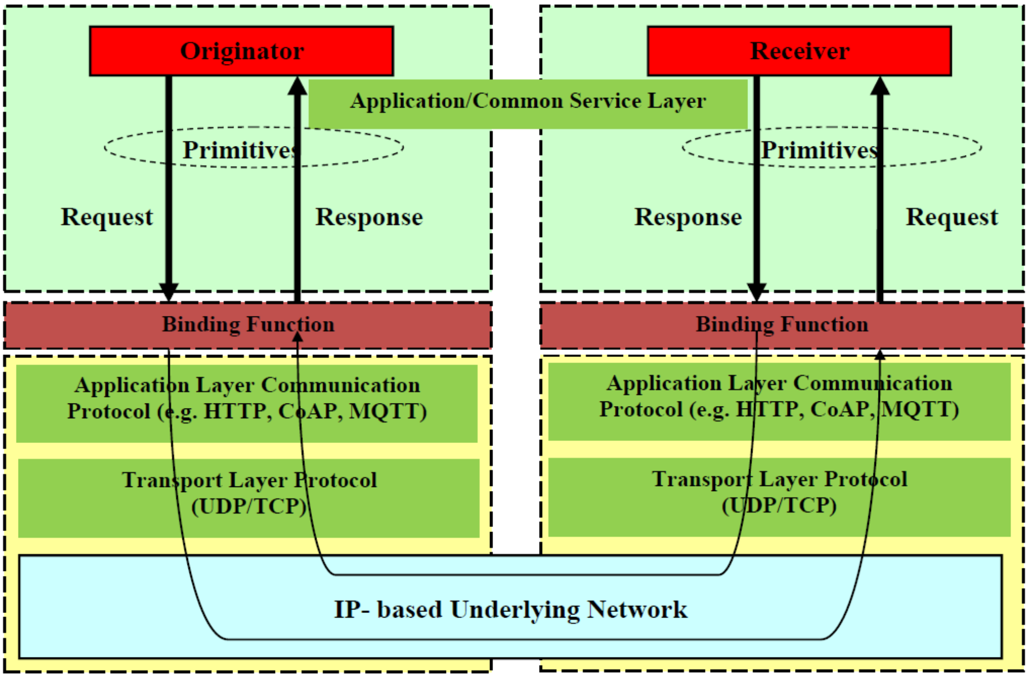
- **rs**: response code: This parameter indicates whether the operation was successful, unsuccessful or is an acknowledgement
- **ri:** Request Identifier. The ri in the Response shall match the ri in the corresponding Request.
- **cn: (conditional)** resource content.

Primitive Mapping

- Primitives are service layer messages transmitted over the Mca and Mcc reference points
- Primitives are data structures that a specific procedure requests or answers in both originator and receiver entities.
- A primitive shall consist of:
  - control part
  - optional content part: user data

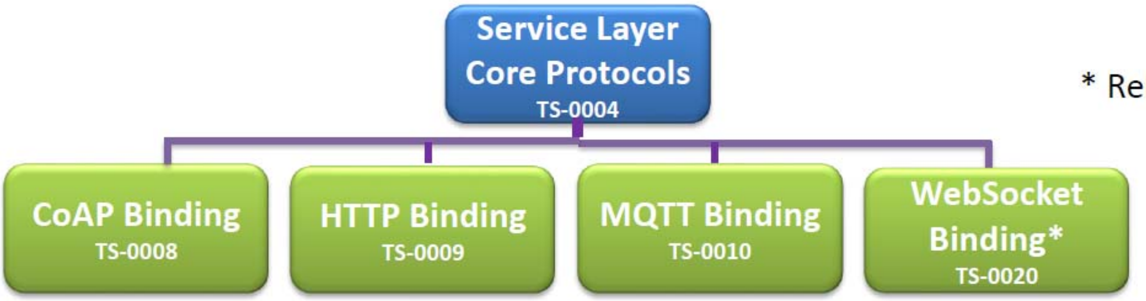
- Designed for Scalability, Extensibility and Efficiency

Primitive Mapping


Originator
Receiver
Application/Common Service Layer
Primitives
Primitives
Request
Response
Response
Request
Binding Function
Binding Function
Application Layer Communication Protocol (e.g. HTTP, CoAP, MQTT)
Application Layer Communication Protocol (e.g. HTTP, CoAP, MQTT)
Transport Layer Protocol (UDP/TCP)
Transport Layer Protocol (UDP/TCP)
IP- based Underlying Network

Source: oneM2M TS-0004

Protocol Binding



Service Layer Core Protocols
TS-0004

CoAP Binding
TS-0008

HTTP Binding
TS-0009

MQTT Binding
TS-0010

WebSocket Binding*
TS-0020

* Release 2

<xxx Resource> Primitives

- xxxCreateRequest
  - xxxCreateResponse
- xxxRetrieveRequest
  - xxxRetrieveResponse
- xxxUpdateRequest
  - xxxUpdateResponse
- xxxDeleteRequest
  - xxxDeleteResponse
- Etc.

HTTP Binding for Primitives

Table C.1: HTTP Method Mapping

Primitive type	HTTP METHOD
xxxRetrieveRequest	GET or, Response to POST (long polling on a communicationChannel)
xxxUpdateRequest	PUT or, Response to POST (long polling on a communicationChannel)
xxxCreateRequest	POST or, Response to POST (long polling on a communicationChannel)
xxxDeleteRequest	DELETE or, Response to POST (long polling on a communicationChannel)
xxxExecRequest	POST (without a body) or, Response to POST (long polling on a communicationChannel)
xxxNotifyRequest	POST (asynchronous notify) or, Response to POST (long polling on a notificationChannel) or, Response to POST (long polling on a communicationChannel)

Source: ETSI TS102.921

Open Source Implementations of oneM2M

LAAS-CNRS OM2M Connecting things

KETI OCEAN

CISCO OPEN DAYLIGHT IotDM

Fraunhofer FOKUS open mtc

oasis

Oasis SI oneM2M Server

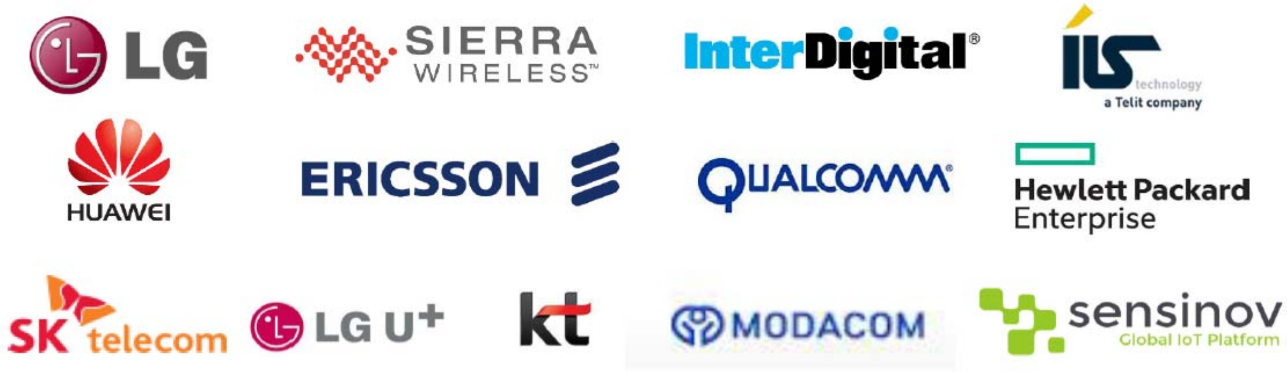
ATIS

OS IoT

one m2m TESTER

oneM2M conformance testing tool project

Commercial Implementations of oneM2M


LG
SIERRA WIRELESS
InterDigital
ILS technology a Telit company
HUAWEI
ERICSSON
QUALCOMM
Hewlett Packard Enterprise
SK telecom
LG U+
kt
MODACOM
sensinov Global IoT Platform

Research based on oneM2M at NCTU

1. Situation Awareness in a Smart Home Environment
2. Gesture-based Control in Smart Home Environment
3. Preventing Misuse of Duplicate Certificates in IoT/M2M Systems
4. Multiple User Activities Recognition in Smart Home
5. Enabling Over-The-Air Provisioning for Wearable Devices
6. A Comparison and Evaluation of Different BLE Connection Methods for Wearable Devices
7. OneM2M-based IoT Protocol Integration
8. Ensuring IoT/M2M System Security under the Limitation of Constrained Gateways
9. Efficient Device Group Management in oneM2M
10. Extending Scalability of IoT/M2M Platforms with Fog Computing
11. Distributed Artificial Intelligence enabled by Fog Networking
12. Enhancing Semantic Discovery in oneM2M with Direct Query

Concluding Remarks (1)

- In this section, we first give an overview of oneM2M Common Service Functions (CSFs) framework.
- We go over each of 12 M2M CSFs.
- Next, a detailed discussion on REST architectural style is given including topics on REST constraints, REST resources, REST interfaces and REST basic operations.
- Then, REST architectural style specific to M2M is elaborated.

Concluding Remarks (2)

- Lastly, the architectural ideas of resource-based M2M communications are explained. Under this architecture, M2M resources are represented as a data model of tree structure.
- We explain this tree structure and the general communication flow used to manipulate this resource tree.
- We also explain protocol binding that maps interface primitives to protocol.
- Finally, available oneM2M system implementations are shown followed by a list of research efforts based on oneM2M.