oneM2M Introduction

W3C WoT Osaka Meeting, May 16, 2017

Yongjing Zhang, zhangyongjing@huawei.com

oneM2M MAS WG Chair,

W3C WoT IG Co-chair

Agenda

- Overview
- Technical Highlights

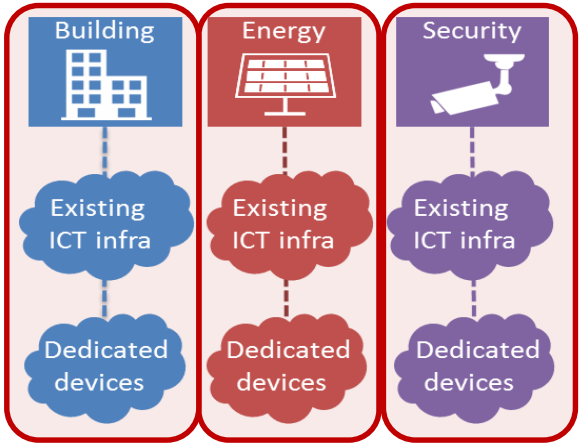
oneM2M Overview

- **Global partnership initiative:** ARIB (Japan), ATIS (N. America), TIA (N. America), CCSA (China), ETSI (Europe), TSDSI (India), TTA (Korea), TTC (Japan)
- **Consolidate standardization of M2M / IoT service functions and APIs**

200+ members organizations

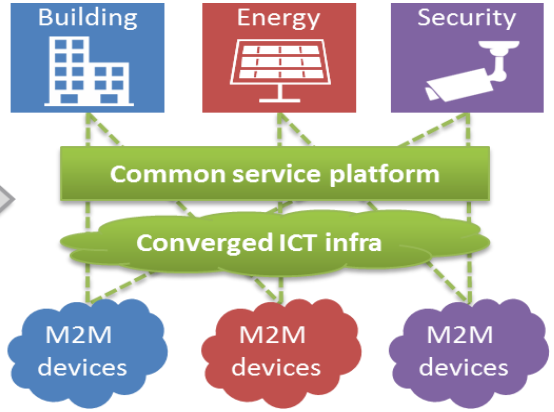
Goal: IoT Cross-Domain Interoperability

- **Standardized Horizontal Service Platform is key enabler for large scale multi-vendor ecosystem with transparent product features and benchmarks, encourages industry investment, and promotes new business models.**



Without oneM2M

- **Highly fragmented market with limited vendor-specific applications**
- **Reinventing the wheel: Same services developed again and again**
- **Each silo contains its own technologies without interoperability**

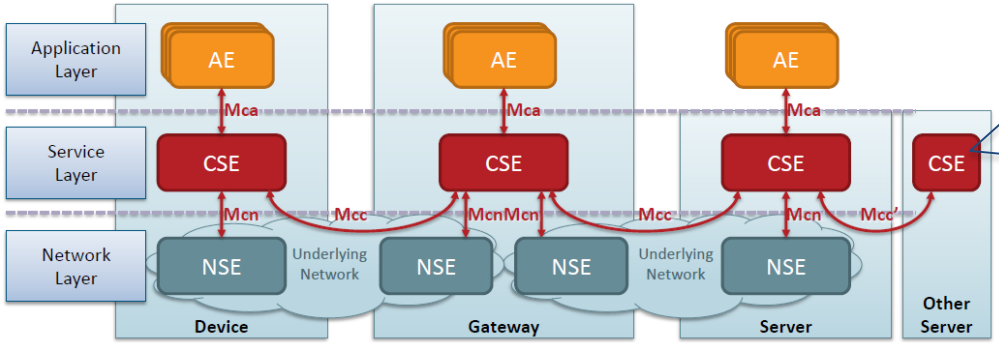


With oneM2M

- **End-to-end platform: common service capabilities layer**
- **Interoperability at the level of communications and data**
- **Seamless interaction between heterogeneous applications and devices**

Functional Architecture

RESTful APIs over Mca & Mcc Reference Points



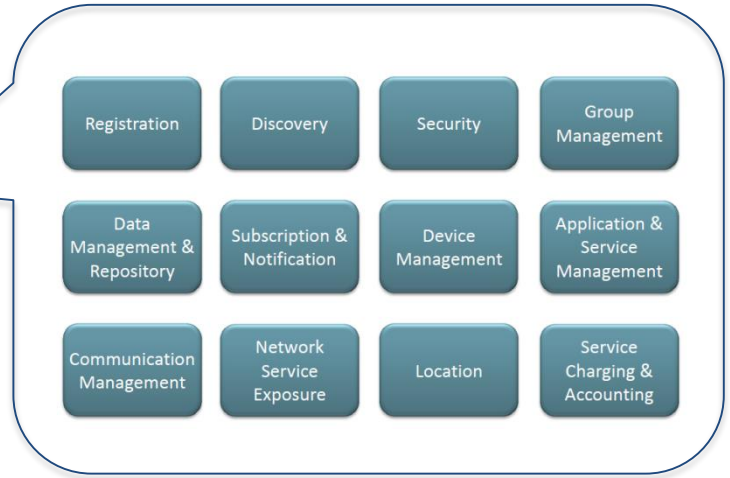
Entities	AE (Application Entity), CSE (Common Services Entity) and NSE (Network Services Entity)
Reference Point	One or more interfaces - Mca, Mcn, Mcc and Mcc'

EXAMPLE REQUEST

```
GET http://provider.net/home/temperature/la
HTTP/1.1
Host: provider.net
X-Orig: /CSE-1234/WeatherApp42
X-M2M-RI: 56398096
Accept: application/vnd.onem2m-res+json
```

EXAMPLE RESPONSE

```
HTTP/1.1 200 OK
X-M2M-RI: 56398096
Content-Type: application/vnd.onem2m-res+json
Content-Length: 94
{"ri":"28375964","cnf":"application/json:0",
"con":"{'timestamp':1413405177000,'value':25.32}"}
```

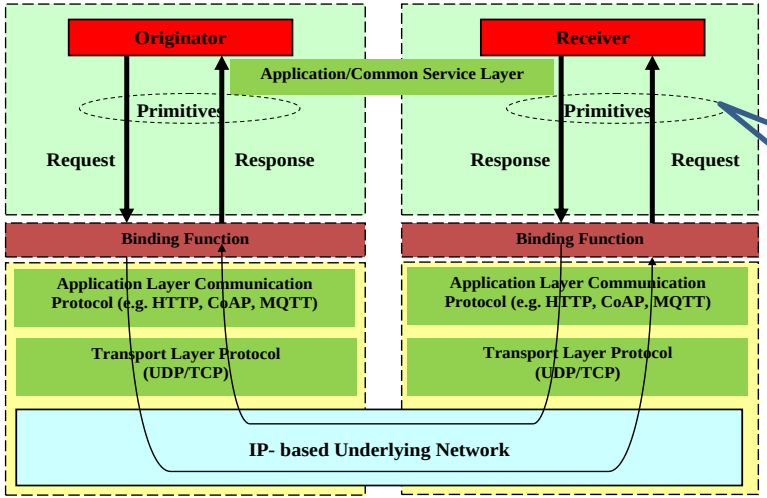



Common Service Functions

Source: " Introduction to oneM2M for the IIC ", IIC Quarterly Meeting, Barcelona, Sept 14th 2015

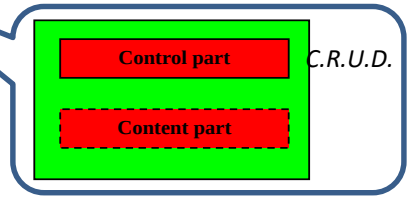
Primitives and Protocol Bindings

- Primitives are common service layer messages exchanged over the Mca, Mcc and Mcc' reference points.
- Primitives are independent of underlying communication protocols and can be mapped to different protocols e.g. HTTP, CoAP MQTT, or WebSocket which use TCP or UDP on the transport layer.

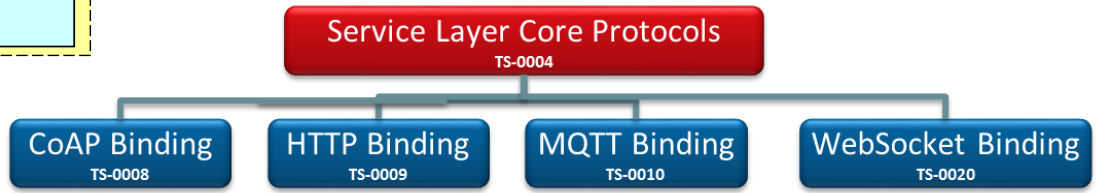


See more details in TS-0004 "Service Layer Core Protocol"

- *The data structure of a primitive consists of two parts:*



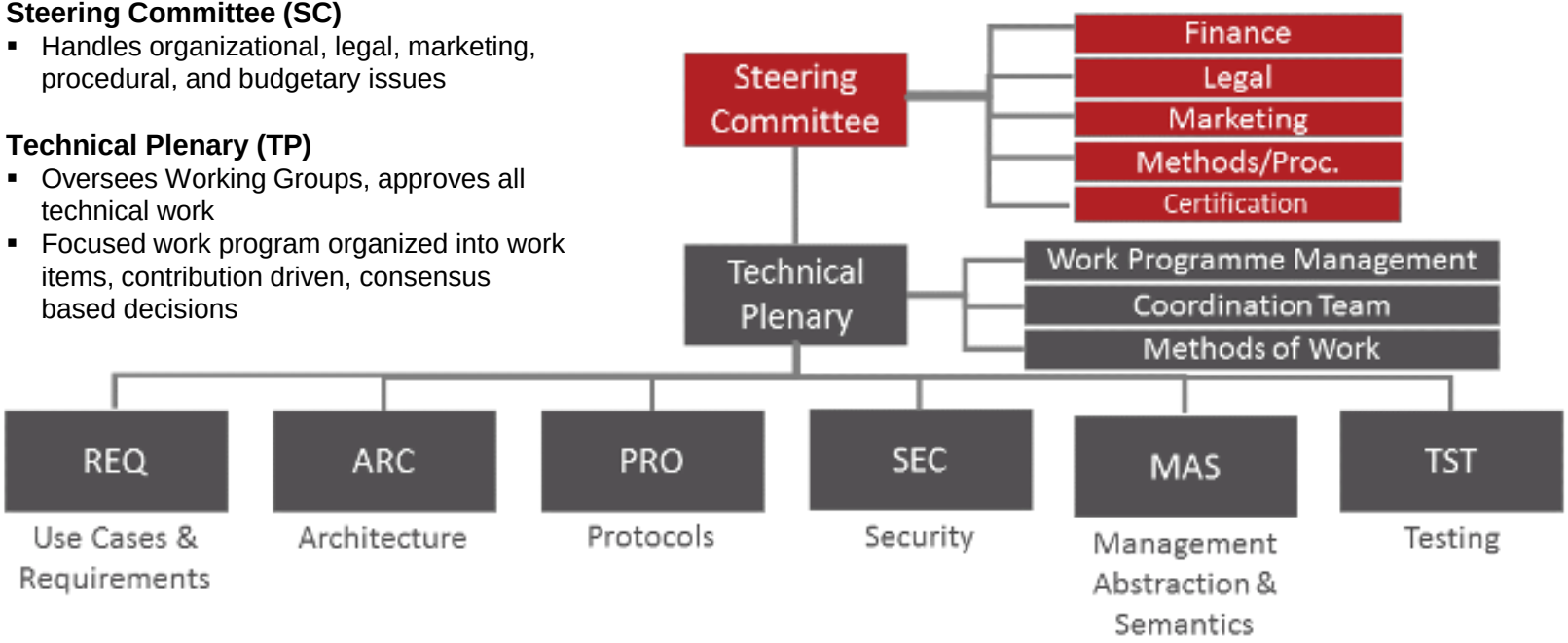
- *Serialization of primitives currently include: JSON, XML and CBOR*



oneM2M Organization

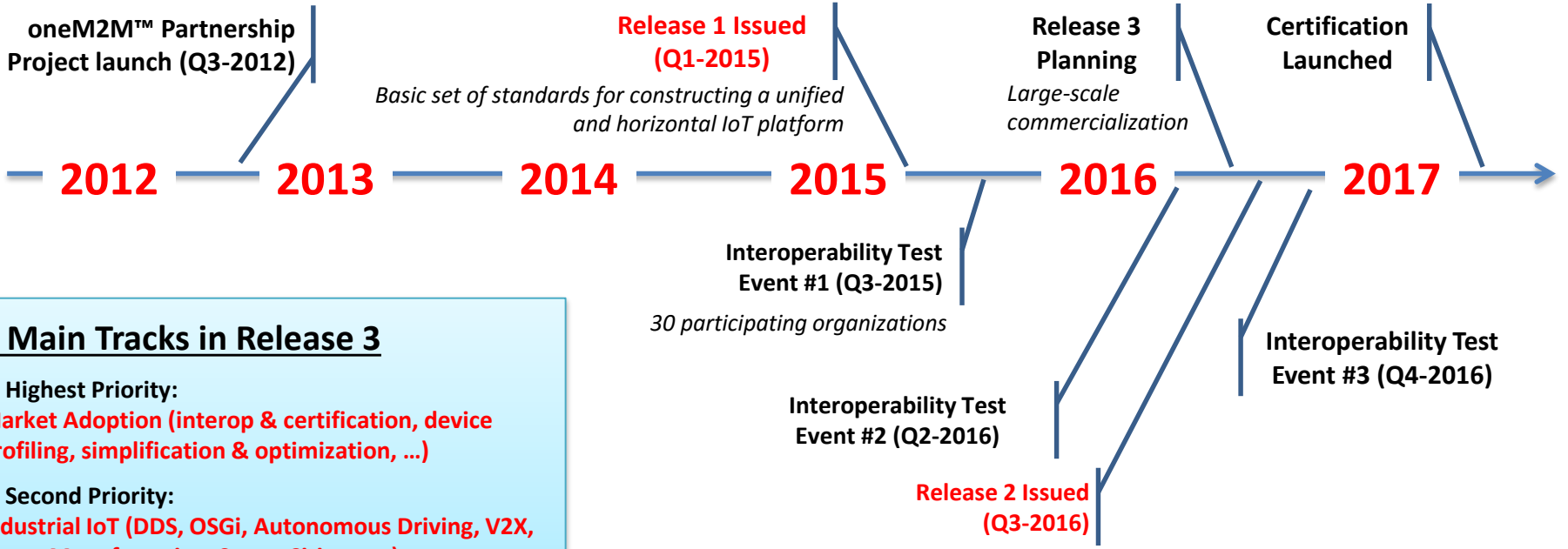
Steering Committee (SC)
- Handles organizational, legal, marketing, procedural, and budgetary issues

Technical Plenary (TP)
- Oversees Working Groups, approves all technical work
- Focused work program organized into work items, contribution driven, consensus based decisions



SOURCE: www.oneM2M.org

oneM2M Project Deliverables



3 Main Tracks in Release 3

1. Highest Priority:
Market Adoption (interop & certification, device profiling, simplification & optimization, …)

2. Second Priority:
Industrial IoT (DDS, OSGi, Autonomous Driving, V2X, Smart Manufacturing, Smart Cities, etc)

3. Third Priority:
Future Looking Topics

oneM2M Rel-2 Ratification (2016.8.30)

Reference	Version	Title
TS 0001	2.10.0	Functional Architecture
TS 0002	2.7.1	Requirements
TS 0003	2.4.1	Security Solutions
TS 0004	2.7.1	Service Layer Core Protocol
TS 0005	2.0.0	Management Enablement (OMA)
TS 0006	2.0.1	Management Enablement (BBF)
TS 0007	2.0.0	Service Components
TS 0009	2.6.1	HTTP Protocol Binding
TS 0010	2.4.1	MQTT Protocol Binding
TS 0011	2.4.1	Common Terminology
TS 0012	2.0.0	oneM2M Base Ontology
TS 0014	2.0.0	LWM2M Interworking
TS 0015	2.0.0	Testing Framework
TS 0020	2.0.0	Websocket Protocol Binding
TS 0021	2.0.0	oneM2M and AllJoyn Interworking
TS 0023	2.0.0	Home Appliances Information Model and Mapping
TS 0024	2.0.0	OIC Interworking

source: http://www.onem2m.org/technical/published-documents

- Inheritance from Rel-1: DM, HTTP/MQTT/CoAP bindings
- Additional protocol binding: WebSocket
- Vertical support extension: home, industrial domain
- Interworking enhancement: AllJoyn, OIC, LWM2M, 3GPP
- Semantic interoperability: annotation, discovery, Base Ontology
- Security enhancement e.g. dynamic authorization
- Testing framework

TR 0001	2.4.1	Use Cases Collection
TR 0007	2.11.1	Study of Abstraction and Semantics Enablements
TR 0008	2.0.0	Security
TR 0012	2.0.0	oneM2M End-to-End Security and Group Authentication
TR 0016	2.0.0	Study of Authorization Architecture for Supporting Heterogeneous Access Control Policies
TR 0017	2.0.0	Home Domain Abstract Information Model
TR 0018	2.0.0	Industrial Domain Enablement
TR 0022	2.0.0	Continuation & integration of HGI Smart Home activities
TR 0024	2.0.0	3GPP_Rel13_IWK

Ongoing work of Rel-3

Extension and continuation from Rel-2, plus new features. Weighted on **Interworking** and **Testing**, **Certification.**

- Interworking related
  - WI-0047 - DDS usage in oneM2M system
  - WI-0048 - OSGi Interworking
  - WI-0052 - LWM2M DM & Interworking Enhancements
  - WI-0056 - Evolution of Proximal IoT Interworking
  - WI-0058 - 3GPP & Cellular IoT Interworking
  - WI-0059 - OPC-UA Interworking
  - WI-0063 - Enhancements on Base Ontology & Generic Interworking
  - WI-0071 - oneM2M and W3C Web of Things Interworking
  - WI-0072 – Modbus interworking
- Vertical
  - WI-0046 - Vehicular domain enablement
  - WI-0064 - Adaptation of oneM2M for Smart City
  - WI-0070 - Disaster Alert Service Enabler
- Development/testing/certification
  - WI-0032 - Conformance Test
  - WI-0051 - Security Functions Conformance Testing
  - WI-0054 - Developers guide series
  - WI-0055 - Product Profiles & Feature Catalog
  - WI-0060 - Interoperability testing Release 2
- Maintenance and small enhancement
  - WI-0015 - oneM2M Use Case Continuation
  - WI-0049 - Rel-1 & 2 Maintenance
  - WI-0050 - Rel-3 Small Technical Enhancements
- Security
  - WI-0019 - Dynamic Authorization for IoT
  - WI-0021 - Secure Environment Abstraction
  - WI-0057 - TEF Interface
  - WI-0061 - Distributed Authorization
  - WI-0065 - Trust management in oneM2M
  - WI-0066 - Decentralized Authentication
  - WI-0067 - UICC Public Key Enhancements
  - WI-0068 - GlobalPlatform Interworking
- Other features and enhancement
  - WI-0030 - M2M Application & Field Domain Component Configuration
  - WI-0031 - Optimized Group-based Operation
  - WI-0034 - Study of re-usable service layer context
  - WI-0035 - Action Triggering
  - WI-0053 - Rel-3 Enhancements on Semantic Support
  - WI-0062 - Service Layer Forwarding
  - WI-0069 - Heterogeneous identification service in oneM2M system

*Note: *Black text: Inherited from R2, Blue text: New in R3*

Strong Implementation Base

Industry-Driven Open Source Implementations

goiot-forum.org

LAAS-CNRS

OM2M Connecting things

CISCO

OPEN DAYLIGHT

IotDM

KETI

OCEAN

Fraunhofer FOKUS

open mtc

atis

OS-IoT

Examples of Commercial Implementations/Demos

SK telecom

LG U+

kt

InterDigital

hp

KEPCO

HUAWEI

SIERRA WIRELESS

sensinov

QUALCOMM

सी-डॉट C-DOT

Creative Convergence ntels

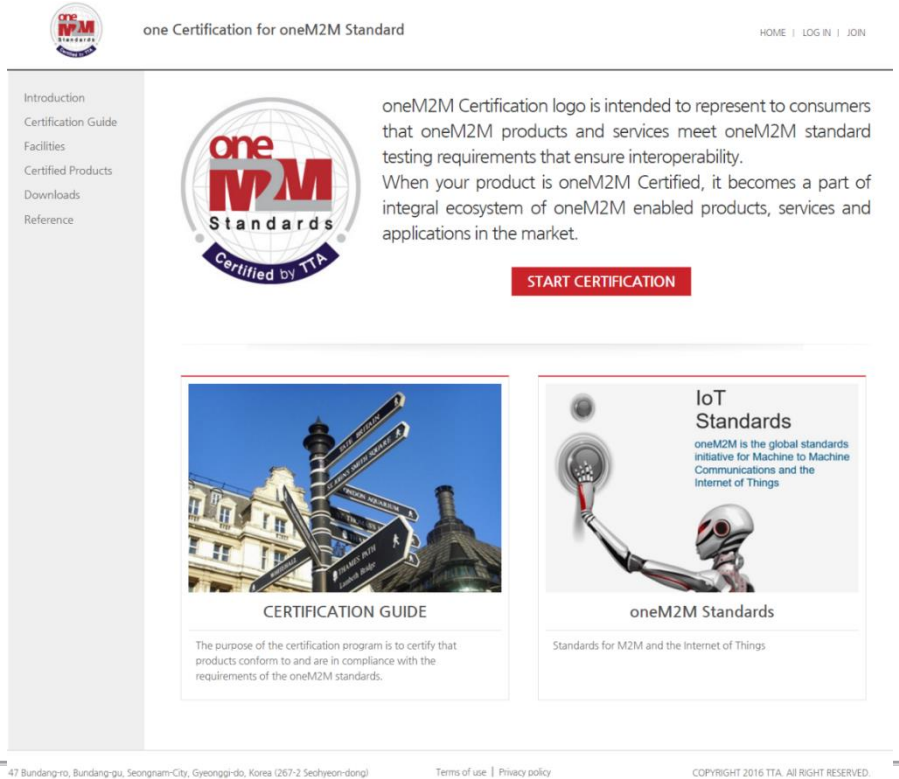
See more: http://www.onem2m.org/news-events/onem2m-deployment-announcements

3 interoperability testing events already (Sept, 2015 France; May, 2016 Korea; Nov 2016, Japan)

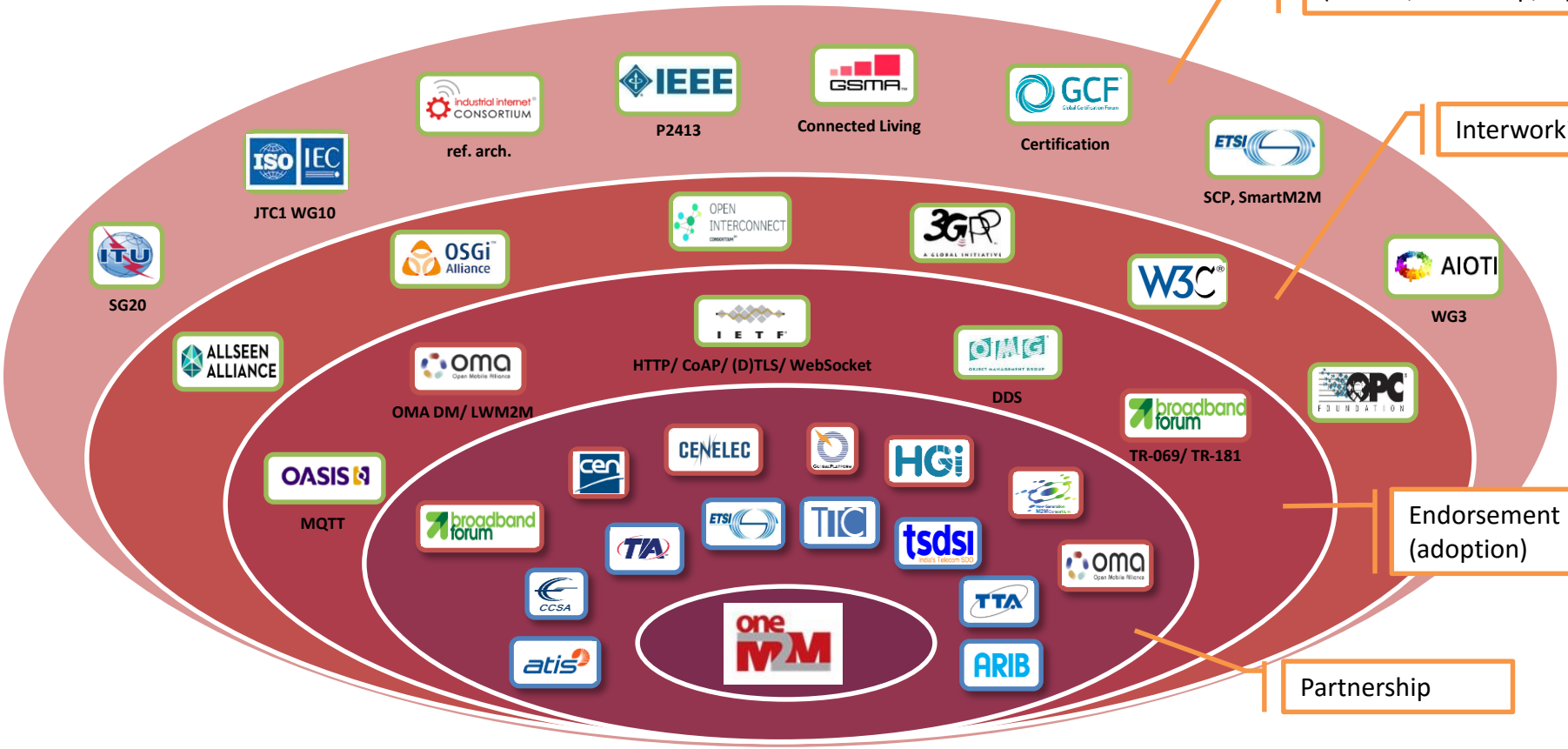
Certification

- oneM2M Certification Program was officially launched at Feb. 9, 2017.
- TTA (Korea) is authorized as the first regional oneM2M CB (Certification Body).
- A Global CB (potentially GCF) would be formally setup in 2018 Q2.

www.oneM2Mcert.com



Industry/Standard Collaboration

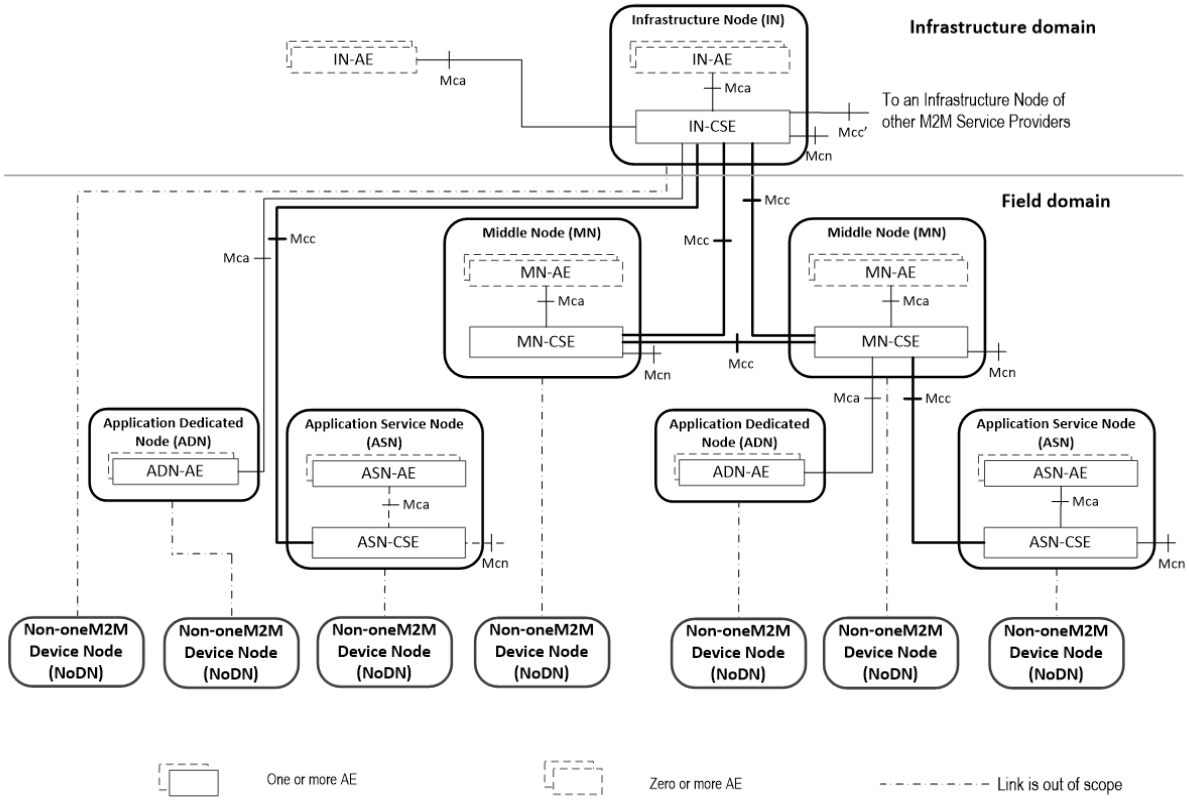


Technical Highlights

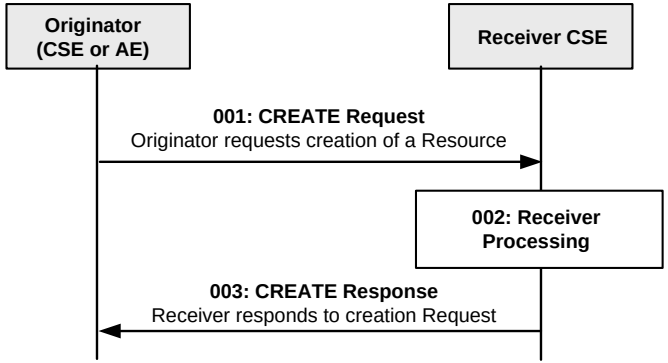
- Architecture Configurations
- Communication Patterns
- Resource Model of a ‘thing’
- Data Management
- Device Management
- Group Management
- Semantics
- Security
- Interworking

Architecture Configurations

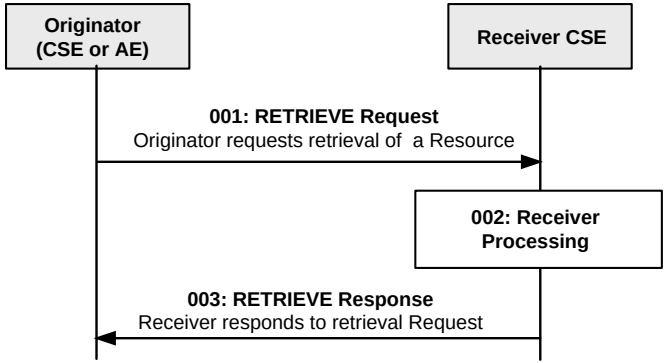
- **Infrastructure Node (IN)**
  - Cloud Platform
- **Middle Node (MN)**
  - Gateway
- **Application Service Node (ASN)**
  - Smart Device (can host local resources)
- **Application Dedicated Node (ADN)**
  - Constrained Device (no resource hosted locally)
- **Non-oneM2M Node (NoDN)**
  - Legacy Device (need interworking)



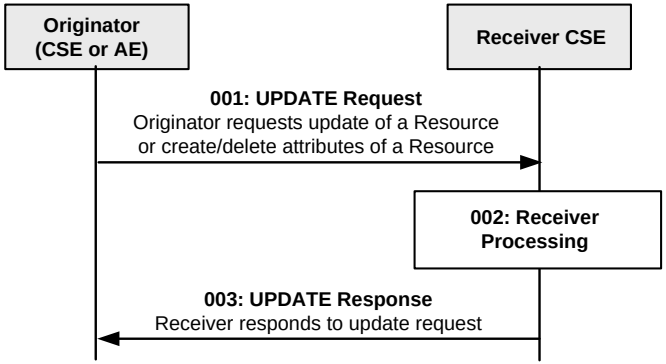
Generic CRUD procedure



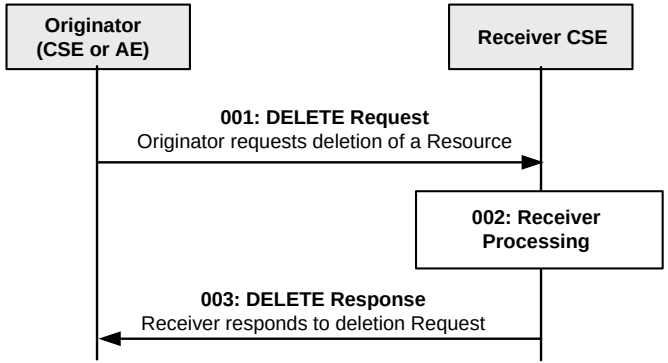
Used to Create a resource



Used to retrieve information from a resource



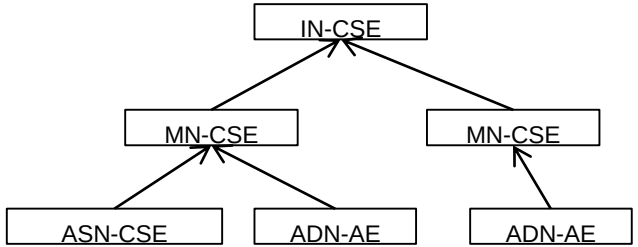
Used to Update a resource



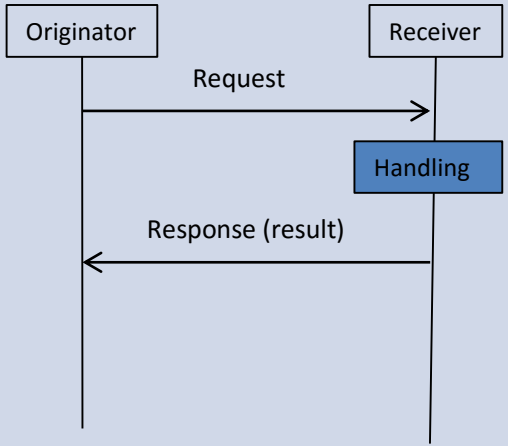
Used to Delete a resource

Communication Models

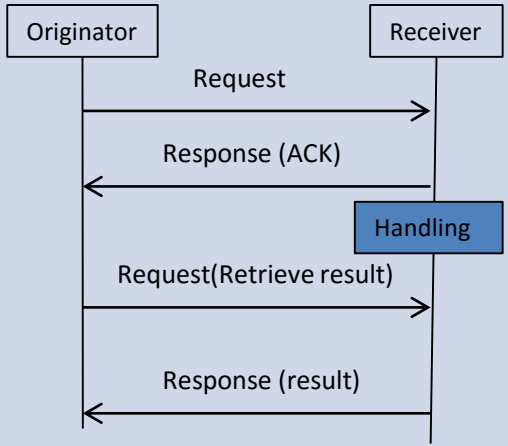
- Every Entity can only register to its upstream Entity
- Every Entity that is the “hub” of several other Entity also serves as a switch of request where IN-CSE serves as the final hub
- All request is delivered to the final destination in a hop-by-hop mode (so far P2P is not supported).



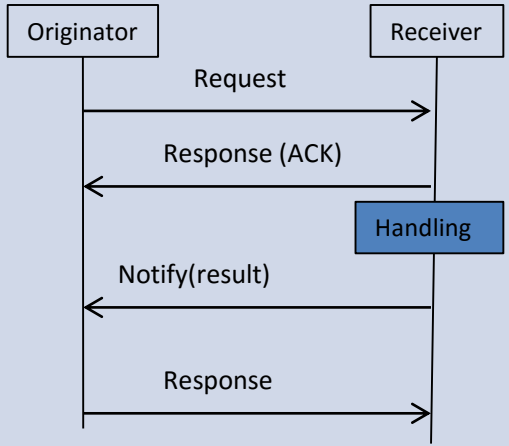
blocking-synchronous



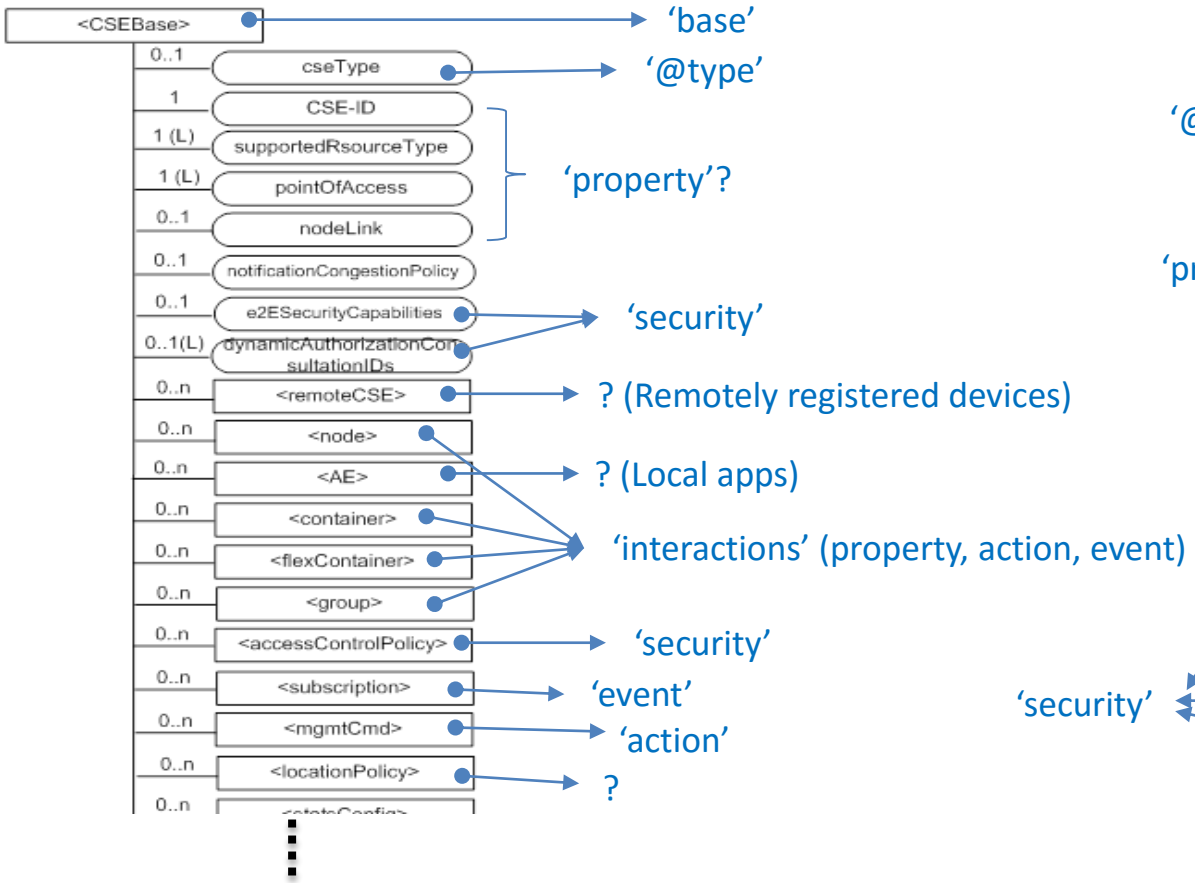
non-blocking synchronous



non-blocking asynchronous



oneM2M Resource Model vs. WoT



Universal Attributes (mandatory)

'@type'

'property'

Attribute Name
resourceType
resourceID
resourceName
parentID
creationTime
lastModifiedTime
expirationTime

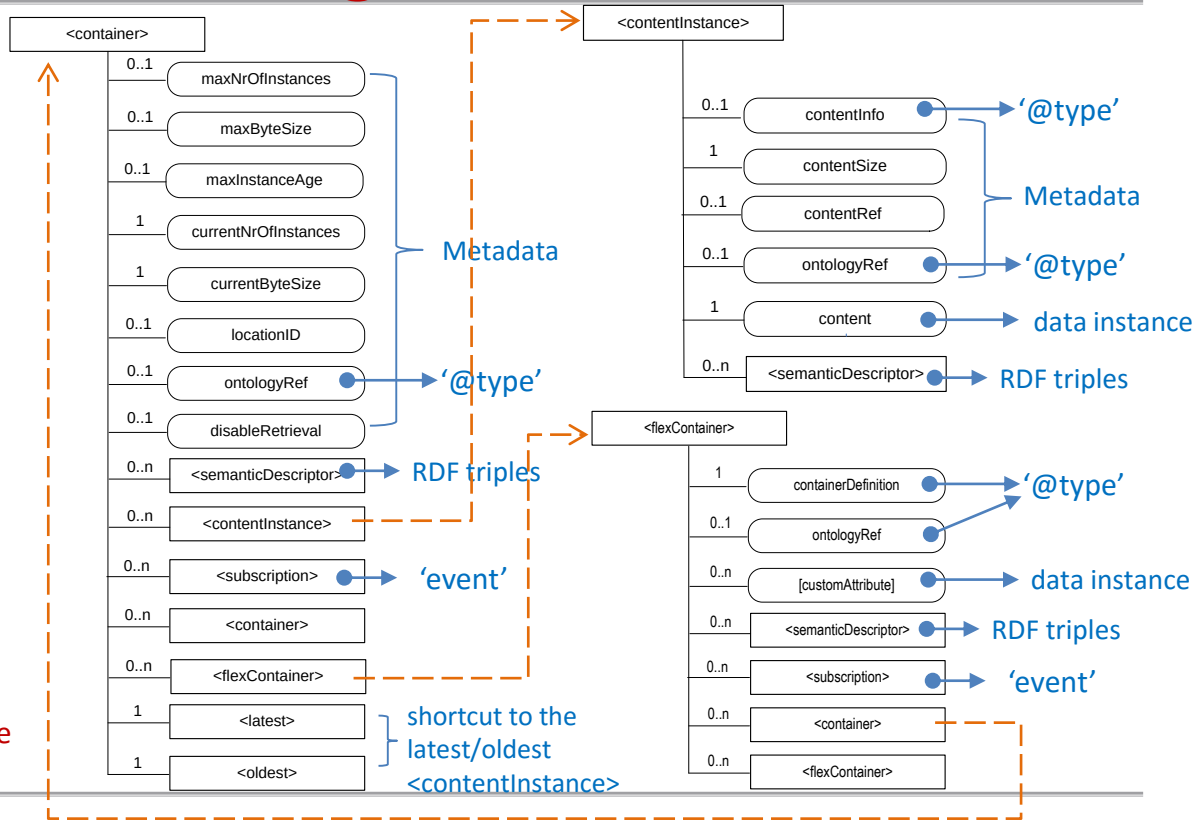
Common Attributes (optional)

'security'

Attribute Name
accessControlPolicyIDs
stateTag
announceTo
announcedAttribute
labels
e2eSecInfo
dynamicAuthorizationConsultationIDs
creator

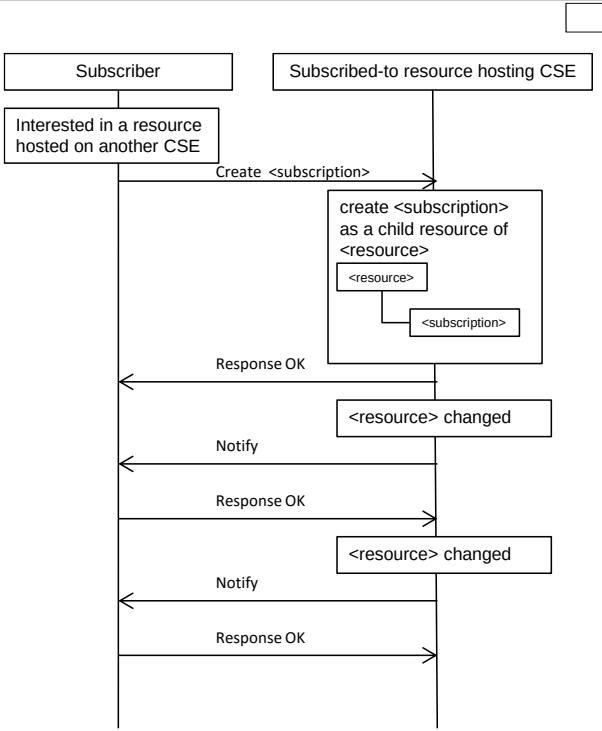
Data Management

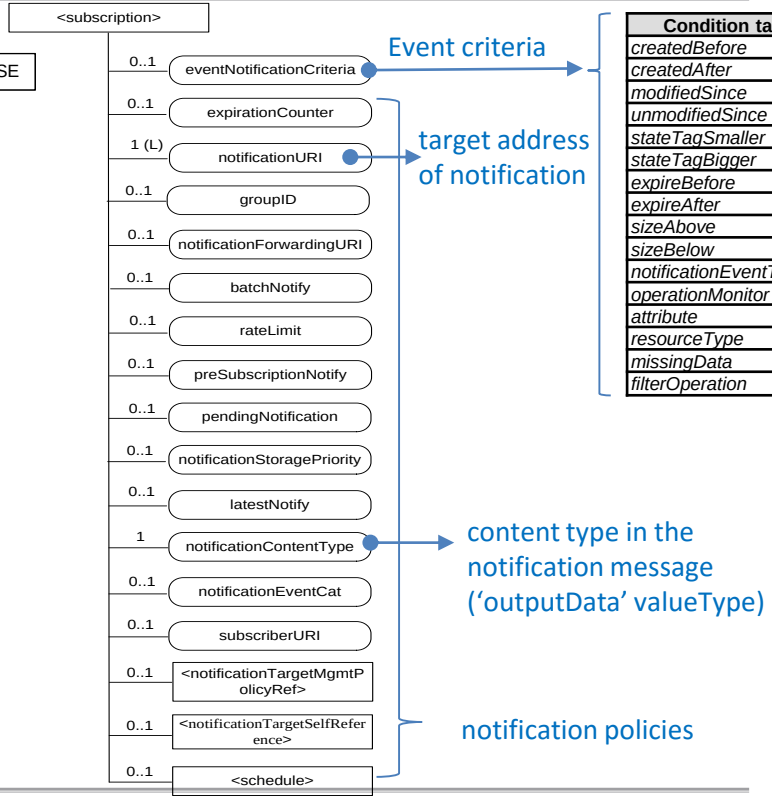
- Different resource types
  - <container> + <contentInstance>
    - multiple instances
    - rich metadata (incl. storage policy)
  - <flexContainer>
    - flatter and simpler structure
    - can be specialized to any data model
  - mixed
- Support hierarchical data model
- Support semantic annotation
- Can represent (depending on implementation context)
  - application data points (‘property’), or
  - service functions (‘action’)
- Eventable
  - by creating <subscription> child-resource



Subscription & Notification (event)

- Subscribe to the change of a resource by creating <subscription> resource, which contains the *notification filterCriteria*, *address* and *policies*.
- Most resource types are subscribable (eventable) by default.
- The notification ('outputData') contains the representation (or partial) of the parent resource being subscribed to.

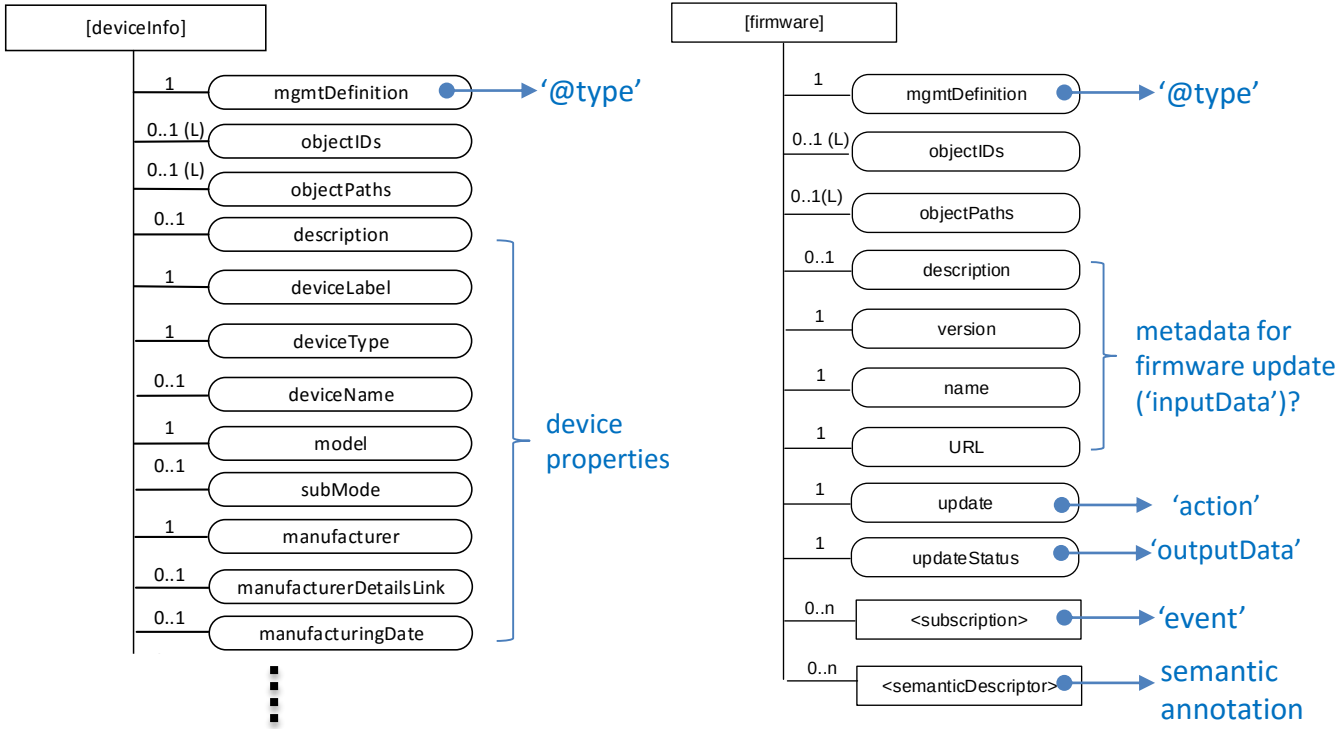




Device Management

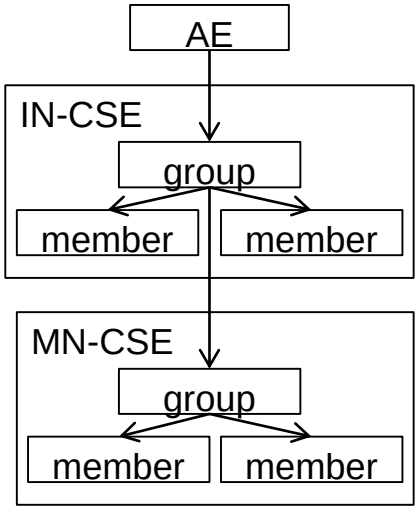
- <mgmtObj> as a template is specialized to individual management resources e.g. [deviceInfo], [firmware]
- Some are actionable, some are not.

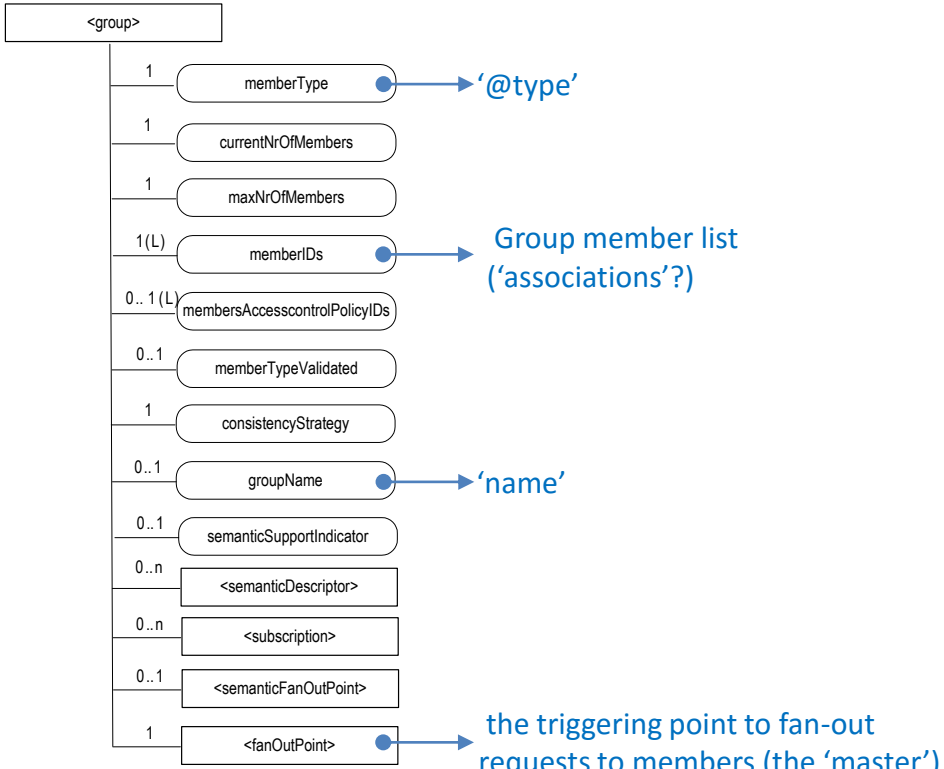
D.2 Resource firmware
D.3 Resource software
D.4 Resource memory
D.5 Resource areaNwkInfo
D.6 Resource areaNwkDeviceInfo
D.7 Resource battery
D.8 Resource deviceInfo
D.9 Resource deviceCapability
D.10 Resource reboot
D.11 Resource eventLog
D.12 Resource cmdhPolicy



Group Management

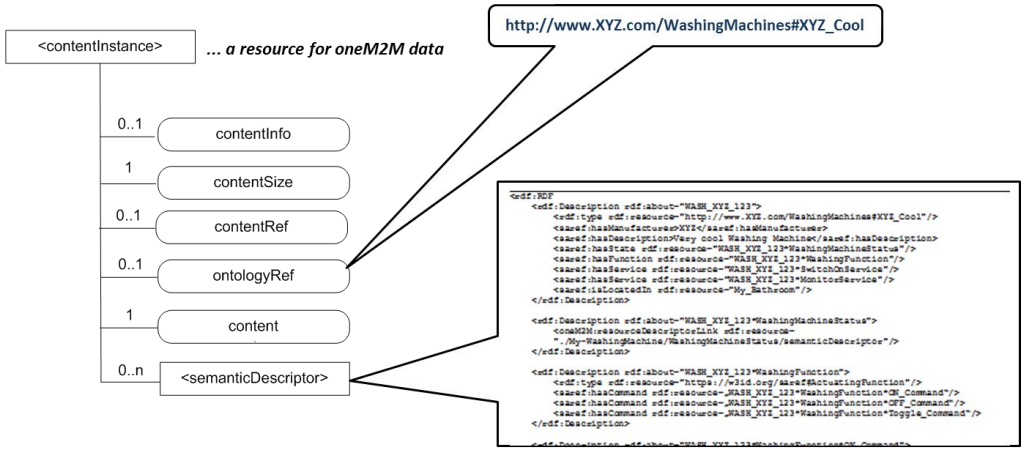
- Distribute requests to and converge responses from multiple devices via a group hosting CSE (device/gateway/platform) to improve communication efficiency





Semantics

- Semantic Annotation
  - Annotate oneM2M data with
    - ✓ *A reference an ontology (= formal description of semantic information) that explains the meaning of the data*

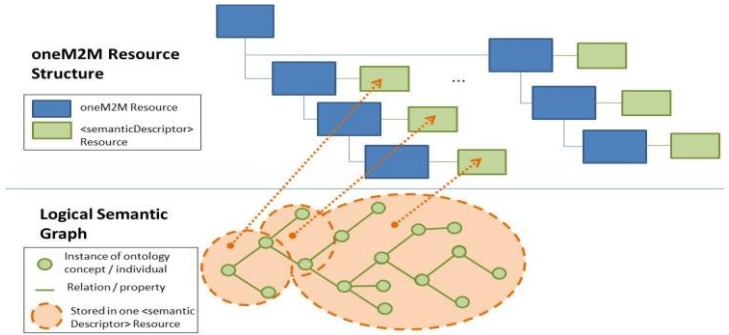


    - ✓ *A description of the data itself and its relation to other data*

… annotations can be done for several oneM2M resource types

- Semantic Discovery/ Query

- Semantic annotation (descriptors) may be distributed in local/remote resource trees.



Example: Discover all resources representing devices that measure temperature.

```
SELECT ?device
    WHERE { ?device rdf:type base:Device .
            ?device base:hasService ?service .
            ?service base:hasFunctionality ?functionality .
            ?functionality rdf:type base:Measuring .
            ?functionality base:refersTo ?aspect .
            ?aspect rdf:type instance:Temperature  }
```

HTTP GET /CSE1234/RCSE78?smf={SPARQL query}

› **More to come: semantic reasoning, mashup, rules, automation …**

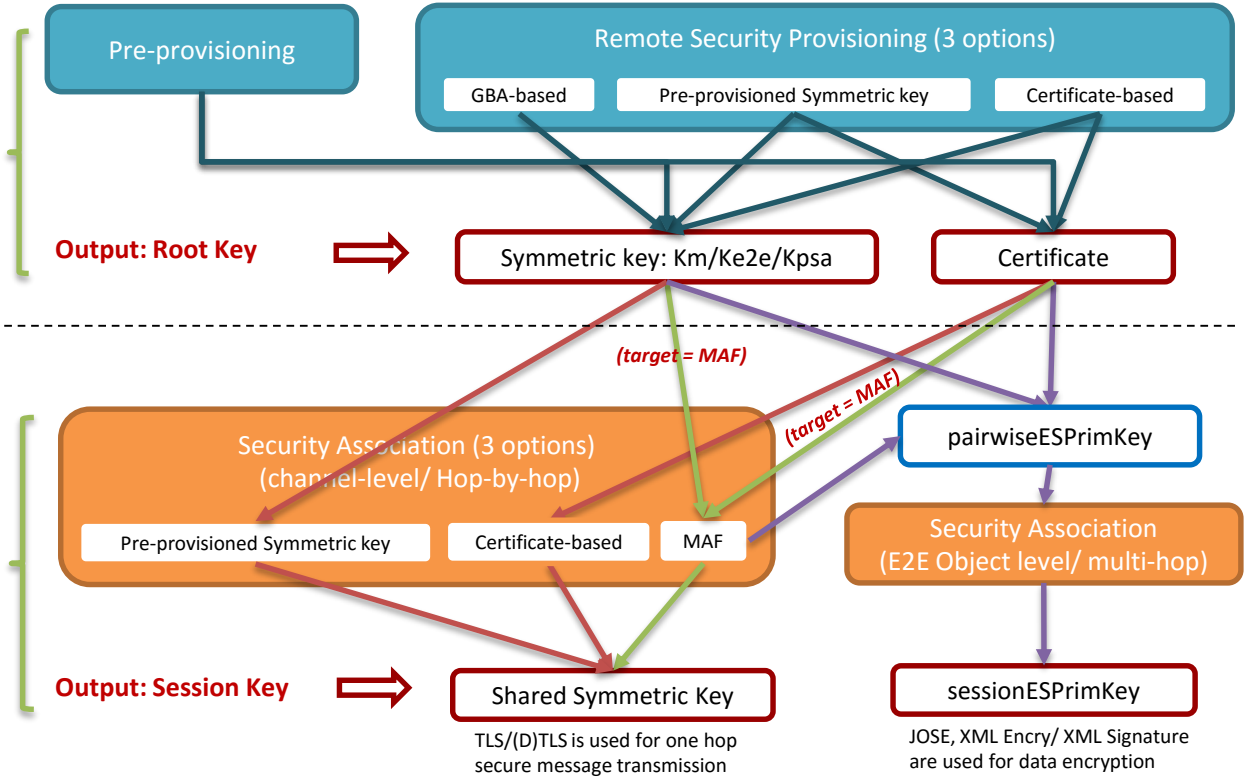
Security: Enrolment & Security Association

Enrolment Phase:

Provision credentials (symmetric key or certificate) to **Enrollee** & **Enrollment Target**

Security Association Phase: establishment of shared security context (session key) between the two **oneM2M entities** to support secure communication

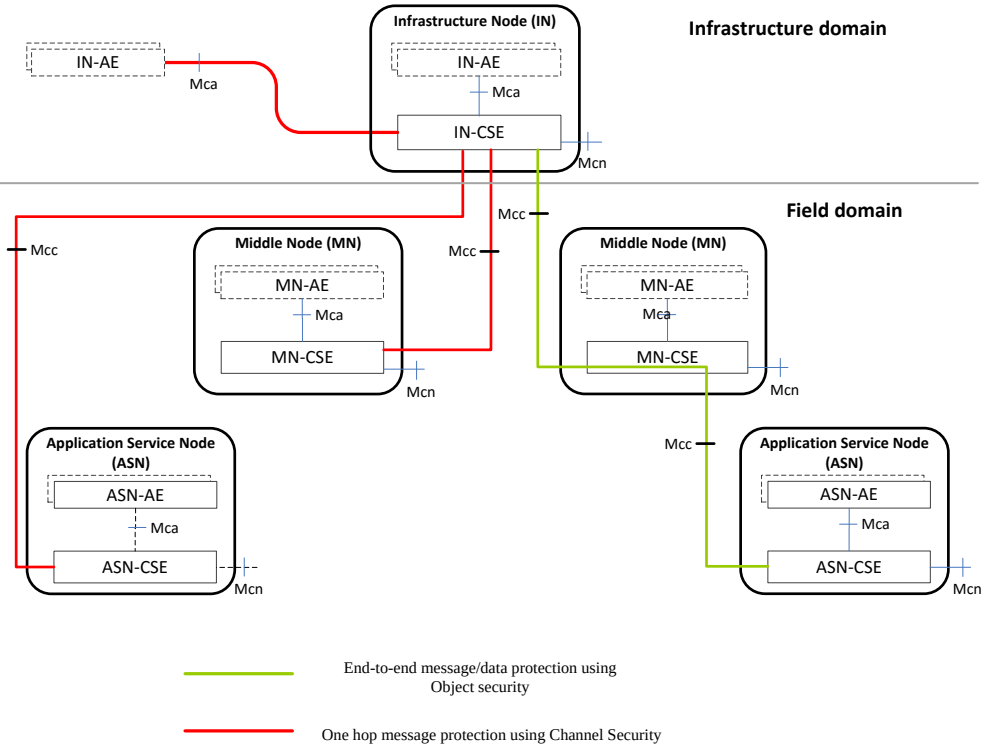
- **Hop-by-hop** channel security
- **End-2-end** object security



**Above Security procedures are all optional.*

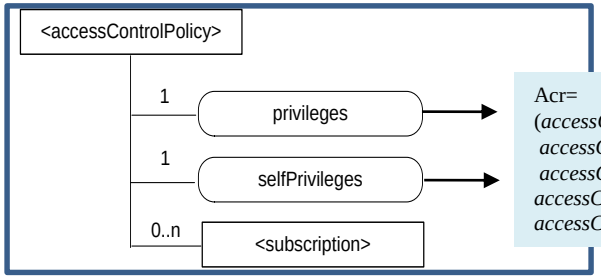
Security: Encryption

- oneM2M supports two encryption mechanism:
  - channel based security: TLS/(D)TLS is used for one hop message transmission
  - object based security: JOSE, XML Encry/XML Signature are used for end-to-end message or data transmission
- Credential used for encryption is generated out of “Security Association” process in the previous slide.

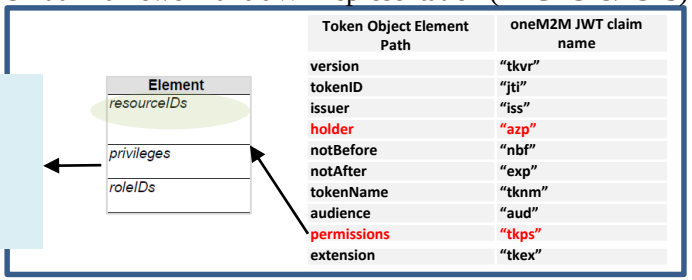


Security: Authorization (Access Control)

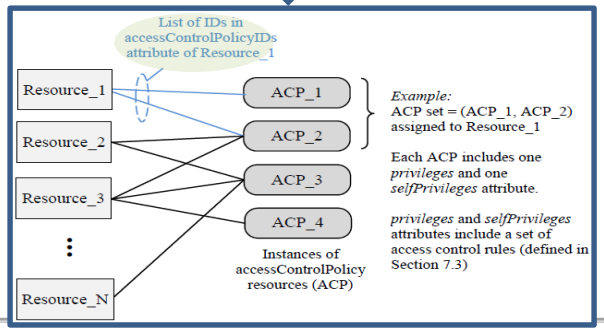
- **ACP-based access control**
  - suitable for relatively static configuration
  - associated with a resource by the accessControlPolicyIDs attribute of the resource
- **Token-based access control**
  - suitable for dynamic authorization
  - associated with a resource by the permissions/resourceIDs element in the Token
  - based on OAuth framework and JWT representation (RFC 7519/7515)

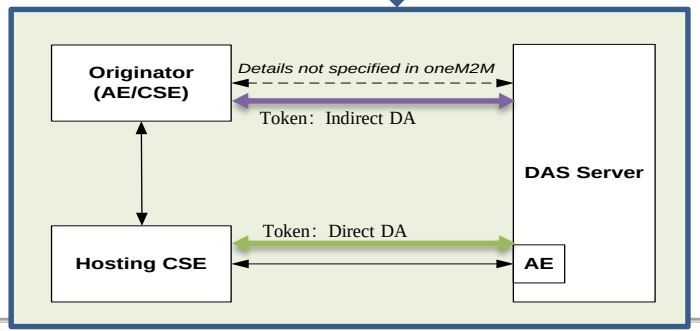


Acr=
(*accessControlOriginators*,
accessControlContexts,
accessControlOperations,
accessControlObjectDetails,
accessControlAuthenticationFlag)

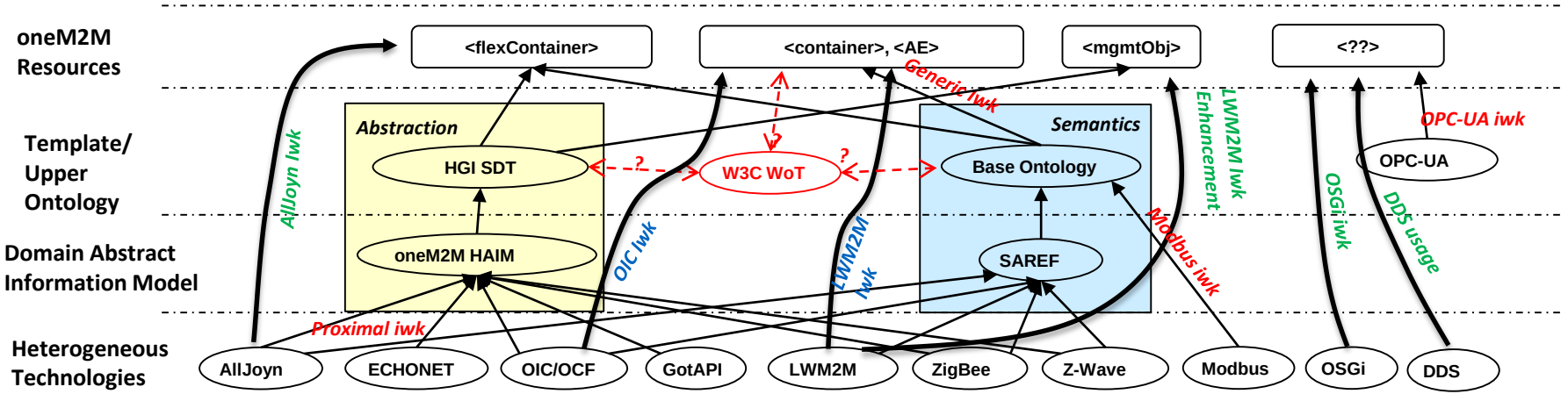


Token Object Element Path	oneM2M JWT claim name
version	"tkvr"
tokenID	"jti"
issuer	"iss"
holder	"azp"
notBefore	"nbf"
notAfter	"exp"
tokenName	"tknm"
audience	"aud"
permissions	"tkps"
extension	"tkex"





oneM2M Interworking Overview



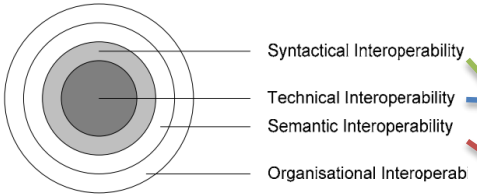


Figure 1: Different levels of interoperability

Source: ETSI IOP Whitepaper 3rd Edition, 2008

To make different standards/technologies working smoothly together

- **Transparent Interworking**: encapsulate the data model of one technology into another (as a pipe)
- **Translucent Interworking:** data model structures are mapped, while semantics/data types are not
- **Semantic Interworking:** mapping not only protocols, but also full data models and semantics

OIC/OCF Interworking

- oneM2M-OIC interworking is based on a ‘transparent’ approach, where OIC/OCF resources are encapsulated (serialized) in a oneM2M <container> resource.
- The oneM2M application (AE) needs to understand the OIC/OCF-native data model inside the <container> to parse and process, while oneM2M CSE doesn’t.
- oneM2M is used as a ‘pipe’ to enable OIC/OCF data exchange over the cloud. Hence the interworking is at the transport level, not semantic level.

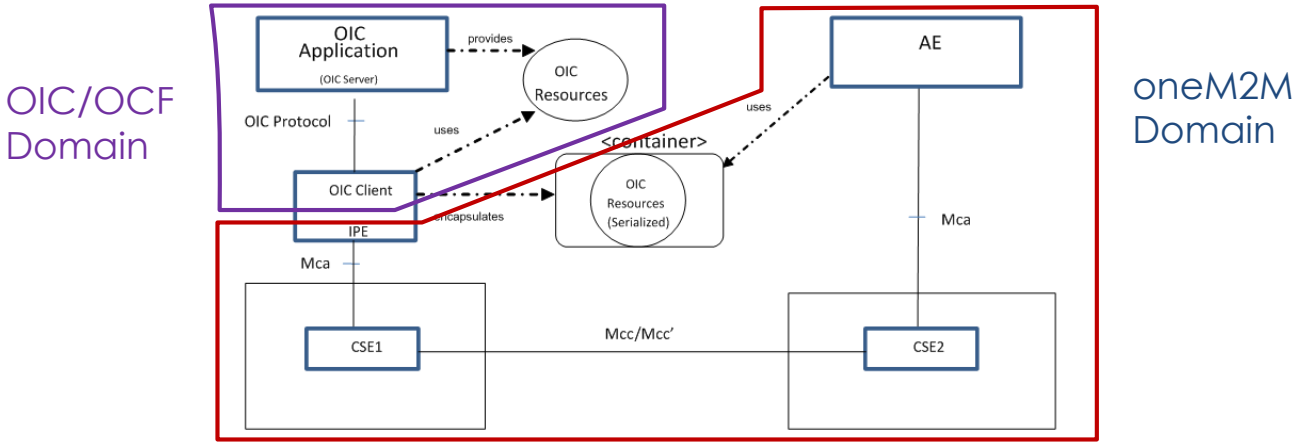
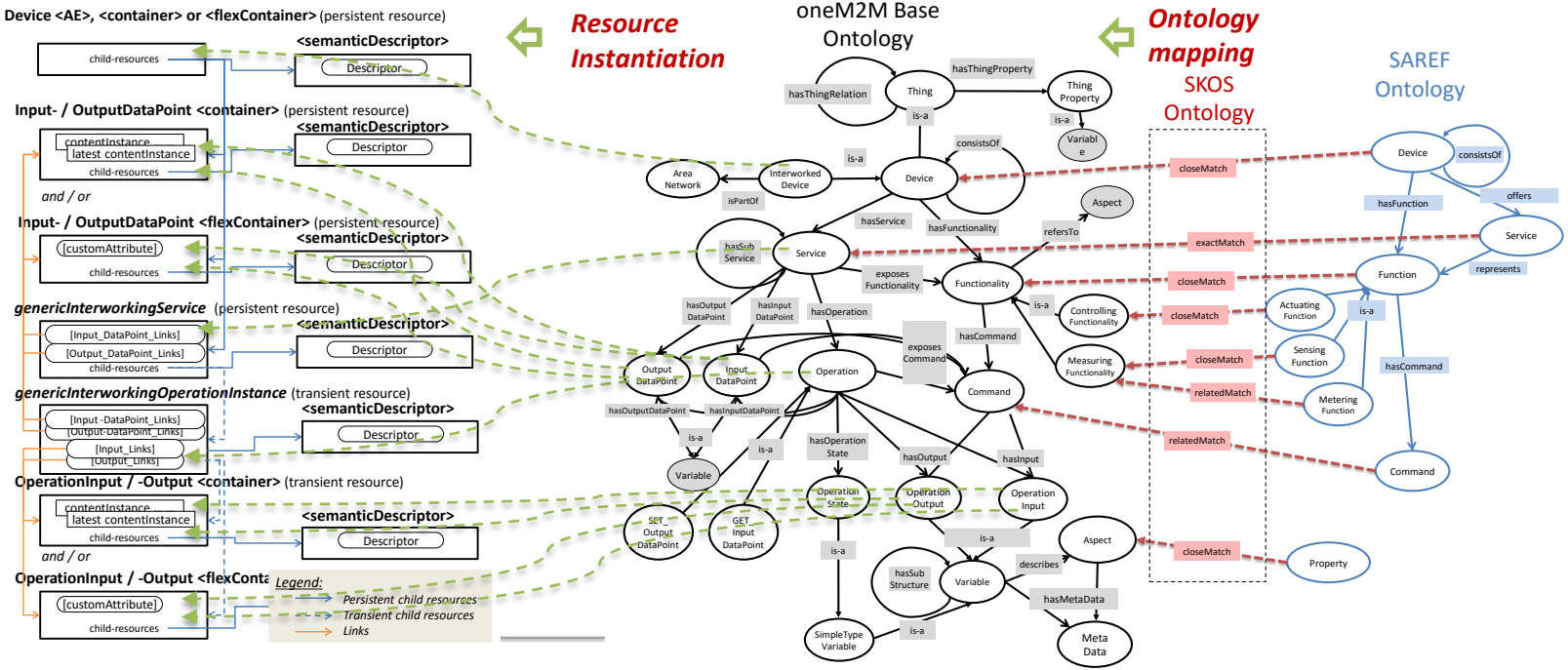


Figure 5.4-1 – OIC Transparent Interworking Function

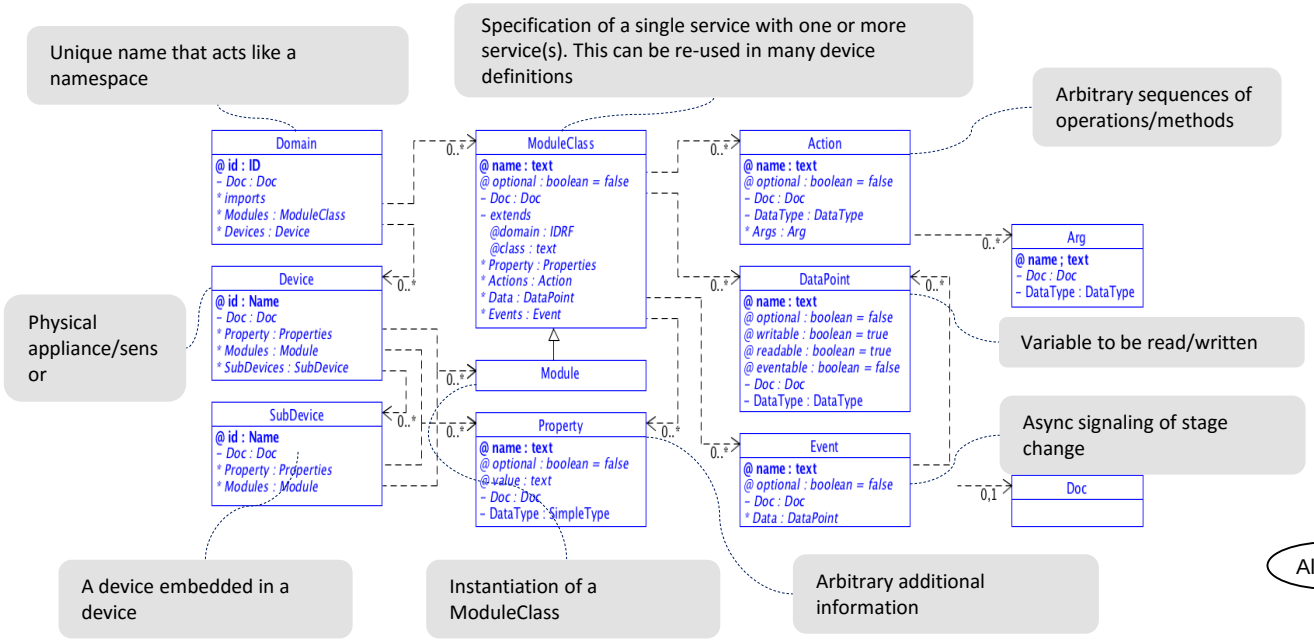
Ontology based Interworking

- oneM2M Base Ontology – the upper ontology serving as the anchor to facilitate/automate the mapping from external system (e.g. SAREF) to oneM2M resource tree.

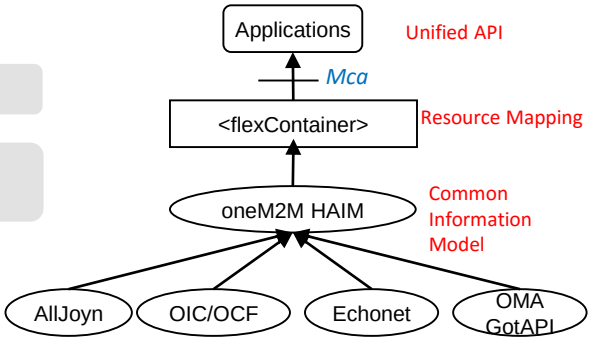


Proximal Interworking via HAIM

HAIM (Home Appliance Information Model) is developed based on HGI SDT (Smart Device Template) 3.0

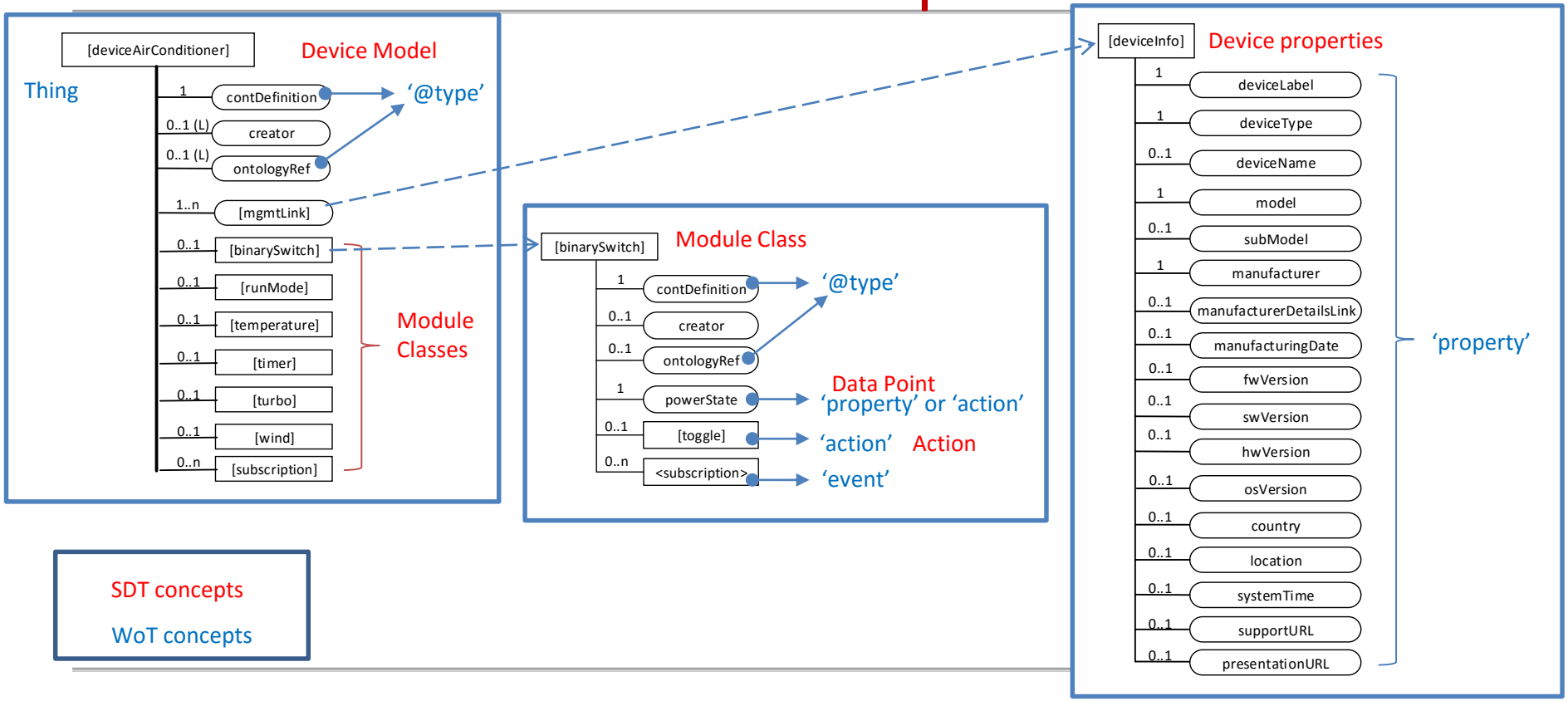


To provided unified API via HAIM, while hiding the heterogeneity of proximal technologies



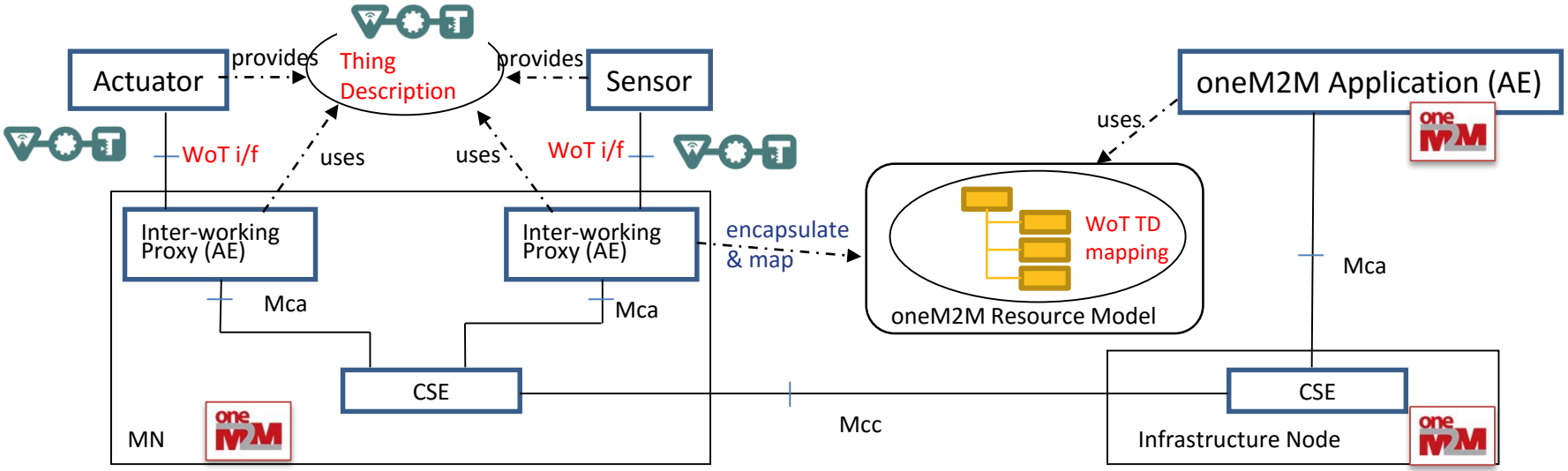
- 30+ Devices (Television, Air conditioner, Oven, …)
- 60+ ModuleClasses (Audio volume, Battery, Binary switch …)

HAIM example



Interworking: WoT→oneM2M

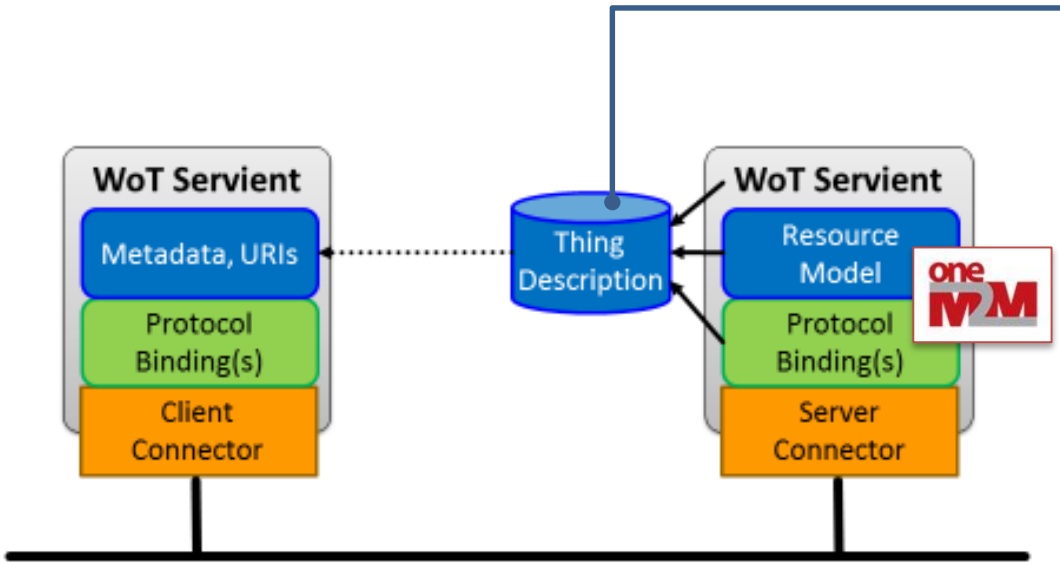
- Exposing the WoT interface (described in TD) to oneM2M systems
  - Benefit: WoT services/data can be consumed by oneM2M applications
  - Question: Does oneM2M Apps need to understand WoT data model at all?



*Note: This slide shows only a preliminary proposal for discussion. Details are FFS (in oneM2M).

Interworking: oneM2M→WoT

- Exposing oneM2M interfaces to WoT systems
  - Benefit: oneM2M services/data can be consumed by WoT Servients
  - Question: is WoT descriptive enough for oneM2M data models and interfaces?




```
{
  "@context": [
    "http://w3c.github.io/wot/w3c-wot-td-context.jsonld",
    { "actuator": "http://example.org/actuator#" }
  ],

  "@type": "Thing",
  "name": "MyLEDThing",

  "uris": [
    "coap://myled.example.com:5683/",
    "http://mything.example.com:8080/myled/"
  ],

  "encodings": ["JSON", "EXI"],
  "security": {
    "cat": "token:jwt",
    "alg": "HS256",
    "as": "https://authority-issuing.example.org"
  },

  "properties": [
    {
      "@type": "actuator:onOffStatus",
      "name": "status",
      "valueType": { "type": "boolean" },
      "writable": true,
      "hrefs": [ "pwr", "status" ]
    }
  ],
```

*Note: This slide shows only a preliminary proposal for discussion. Details are FFS.

Thank you!

Annex: oneM2M specification walkthrough