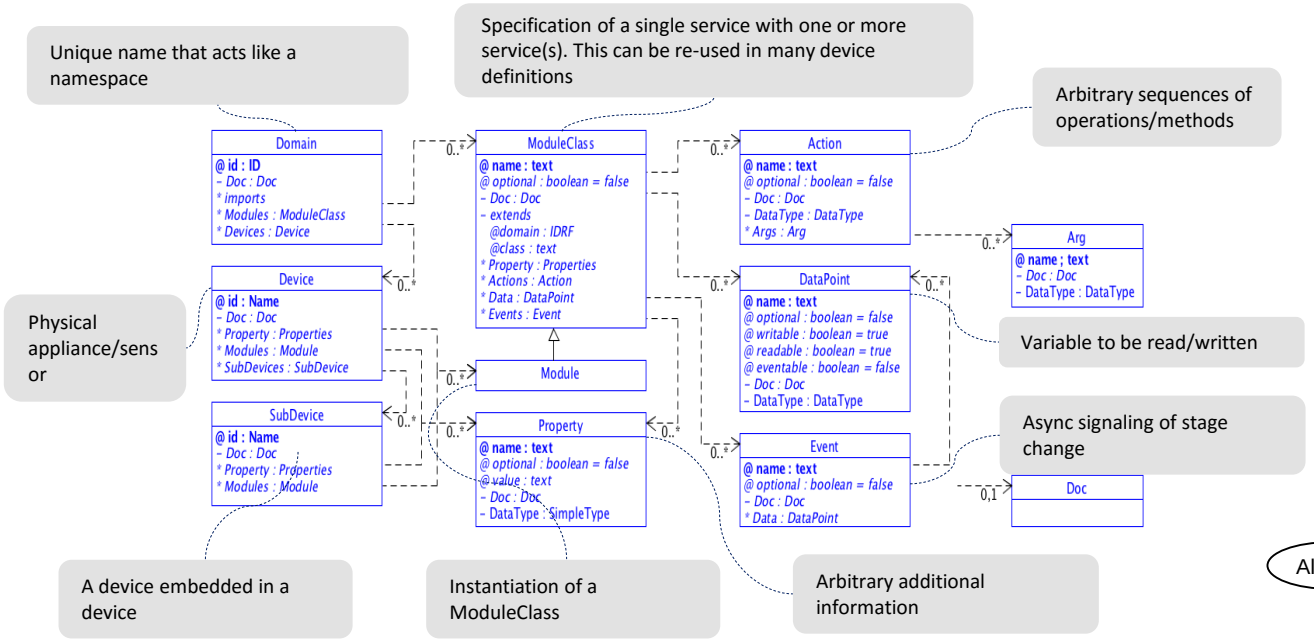
oneM2M HAIM – WoT TD mapping discussion

W3C WoT Osaka Meeting, May 16, 2017

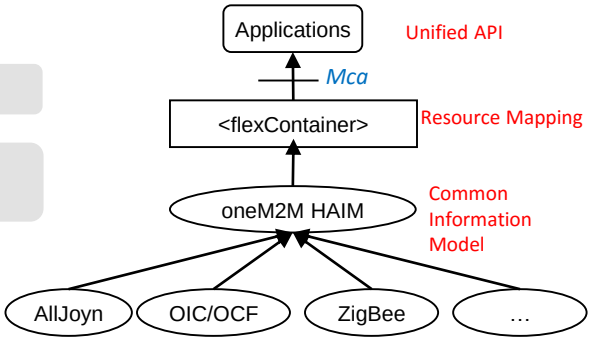
Yongjing Zhang, Zhangyongjing@Huawei.com

Proximal Interworking via HAIM

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

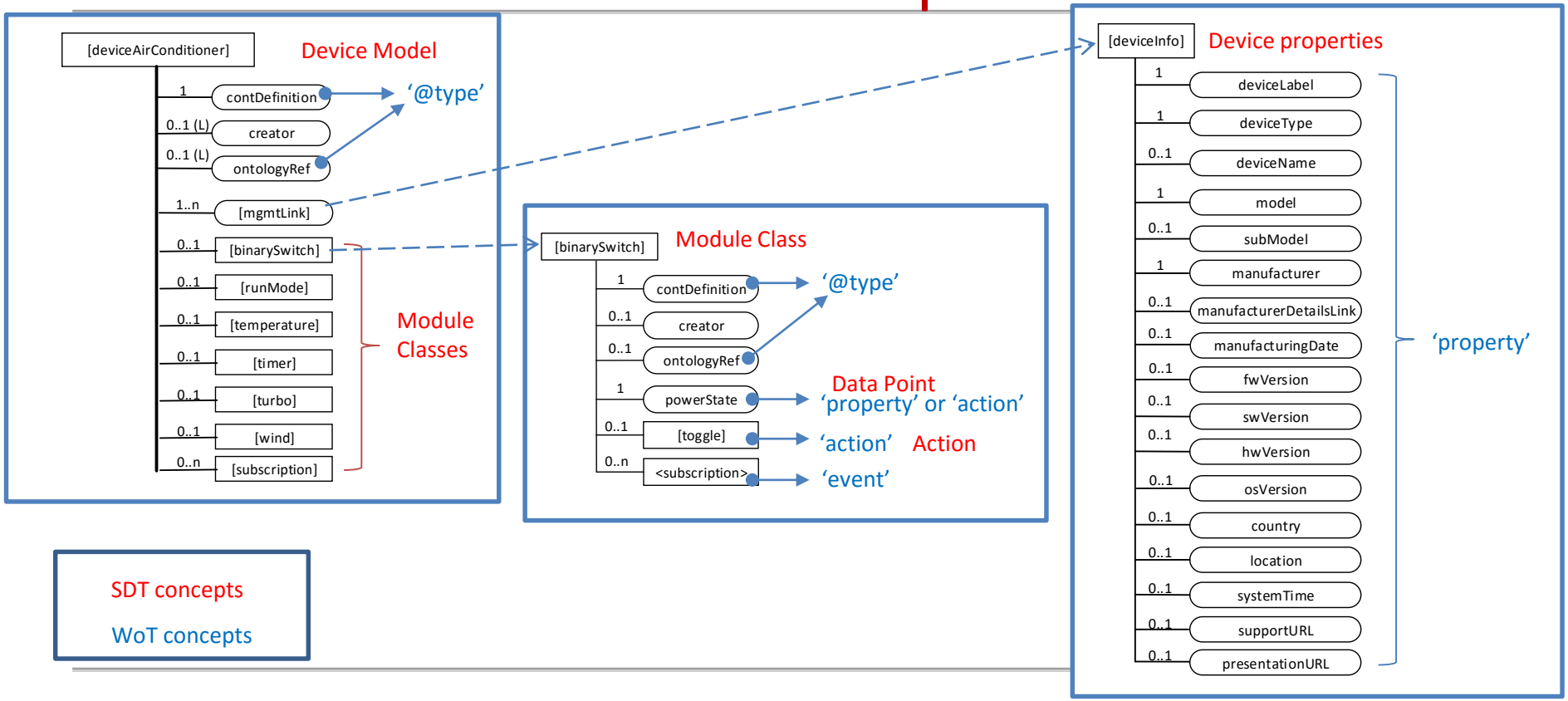


To provided unified API via HAIM, while hiding the heterogeneity of proximal technologies



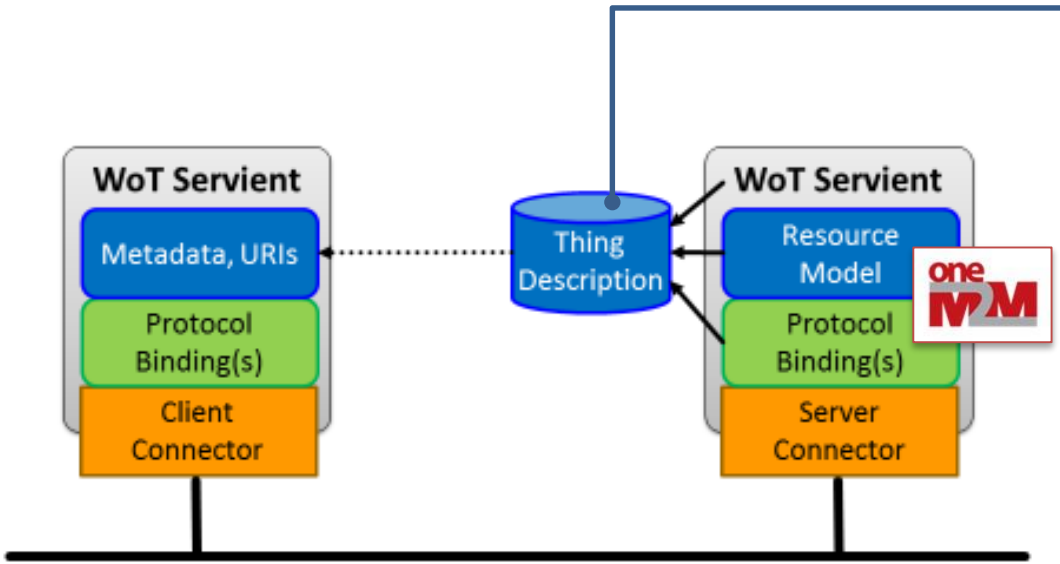
- 30+ Devices (Television, Air conditioner, Oven, …)
- 60+ ModuleClasses (Audio volume, Battery, Binary switch …)

HAIM example



Interworking: oneM2M→ WoT

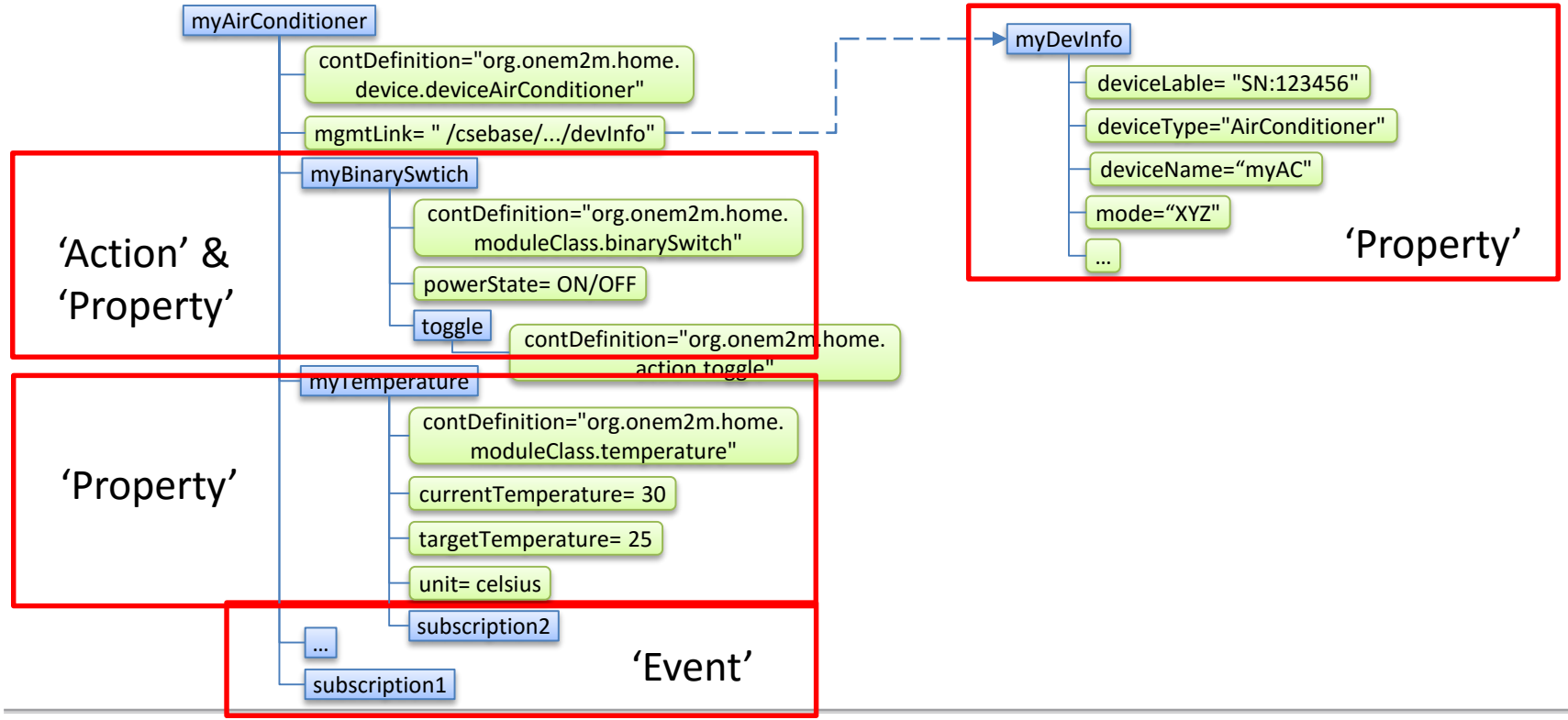
- Exposing oneM2M interfaces to WoT systems
  - Benefit: oneM2M services/data can be consumed by WoT Servients
  - Question: is WoT descriptive enough for oneM2M data models and interfaces?



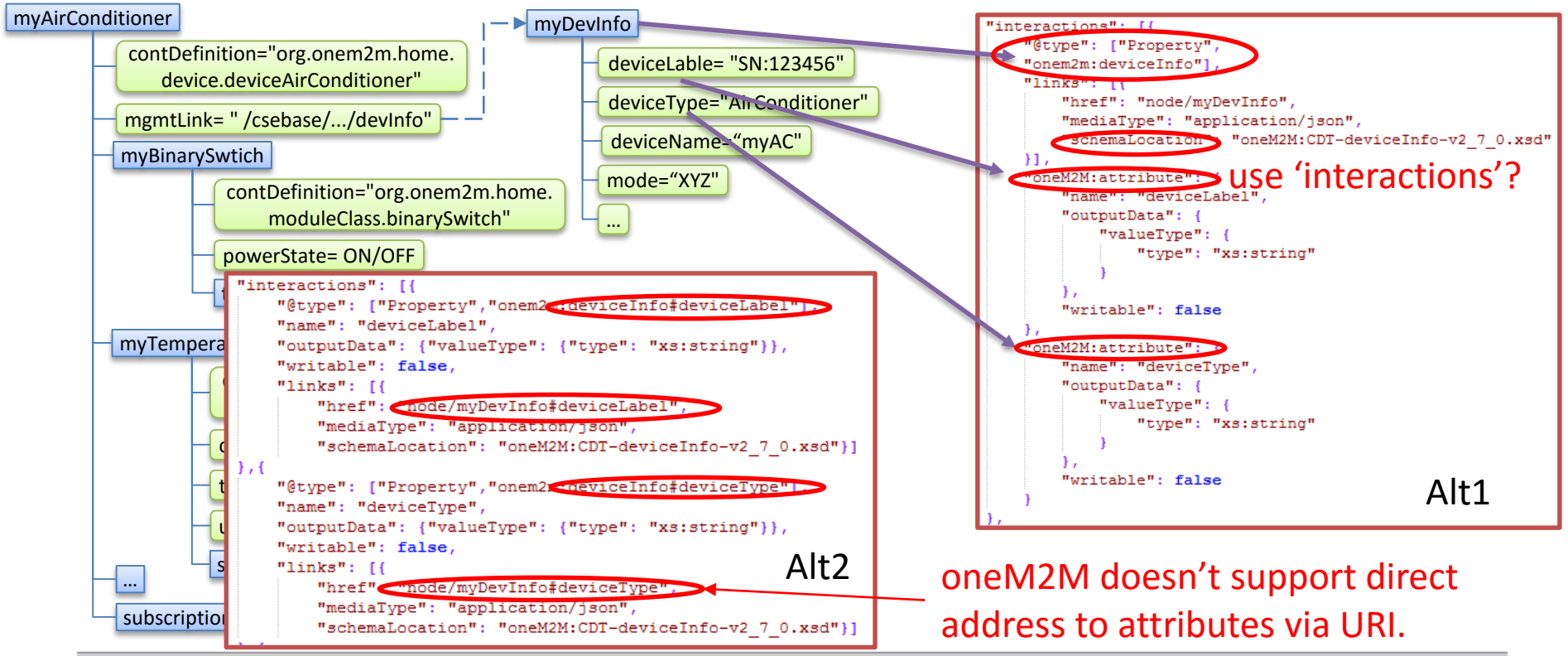

```
{
  "@context": [
    "http://w3c.github.io/wot/w3c-wot-td-context.jsonld",
    { "actuator": "http://example.org/actuator#" }
  ],

  "@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.

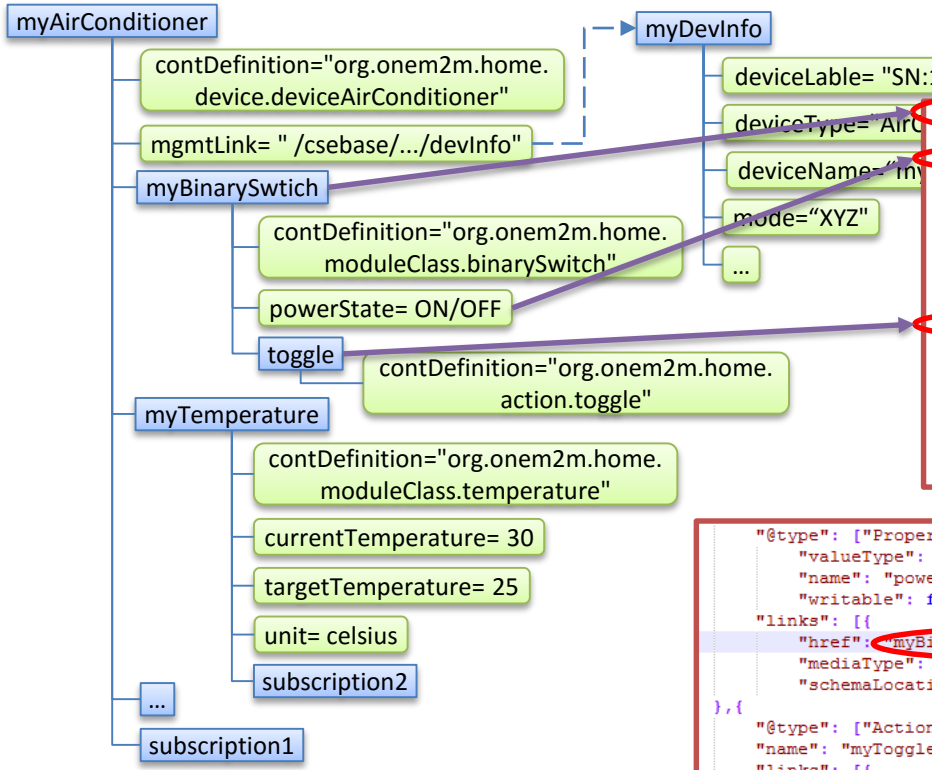
An exemplary mapping to WoT (1/5)



An exemplary mapping to WoT (2/5)



An exemplary mapping to WoT (3/5)



How to describe hierarchical actions?
what about triggering method?

use 'interactions'?

```
"@type": ["onem2m:binarySwitch","haim:org.onem2m.home.moduleClass.binarySwitch"],
"name": "myBinarySwitch",
"oneM2M:attribute": {"valueType": {"type": "xs:boolean"},
    "name": "powerState",
    "writable": false},
"links": [{
    "href": "myBinarySwitch",
    "mediaType": "application/json",
    "schemaLocation": "oneM2M:HD-mod-binarySwitch-v3_0_0.xsd"
}]
"oneM2M:moduleClass": {
    "@type": ["Action","haim:org.onem2m.home.action.toggle"],
    "name": "myToggle",
    "links": [{
        "href": "myBinarySwitch/myToggle",
        "method": "PUT",
        "mediaType": "application/json",
        "schemaLocation": "oneM2M:HD-act-toggle-v3_0_0.xsd"}]},
```

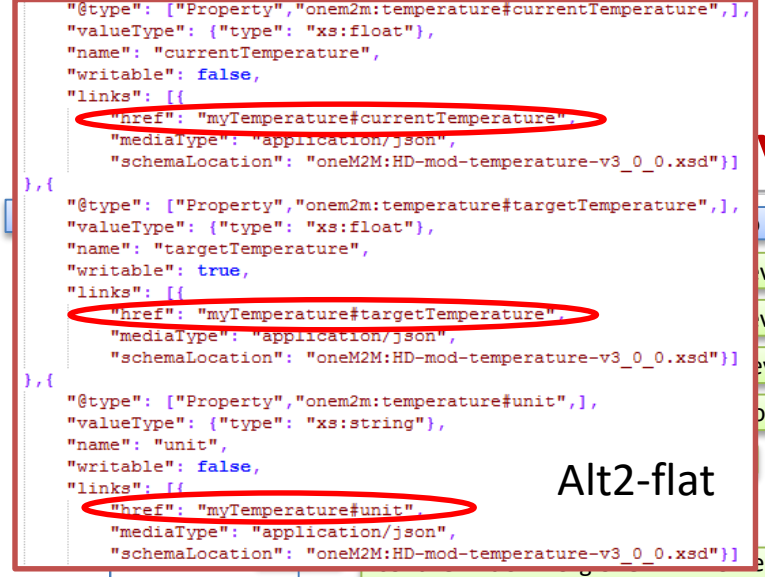
Alt1

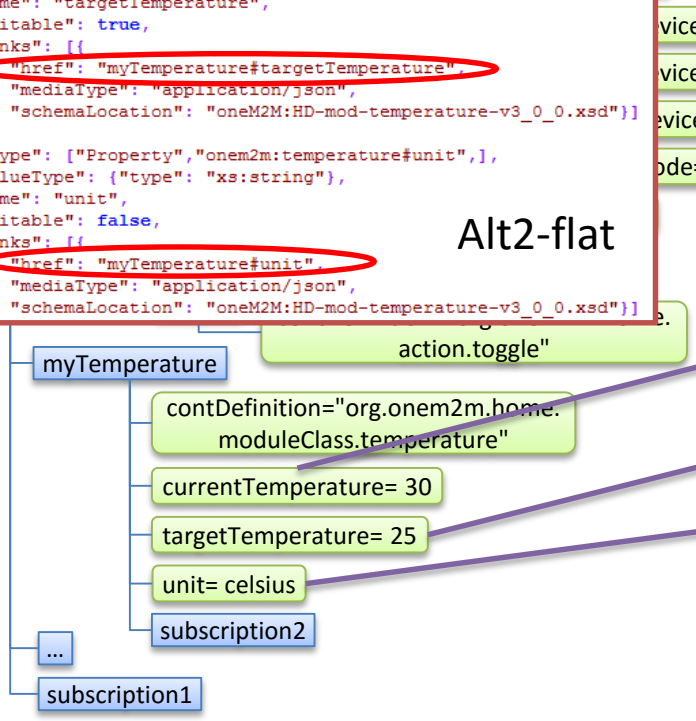
```
"@type": ["Property","onem2m:binarySwitch#powerState"],
    "valueType": {"type": "xs:boolean"},
    "name": "powerState",
    "writable": false,
"links": [{
    "href": "myBinarySwitch#powerState",
    "mediaType": "application/json",
    "schemaLocation": "oneM2M:HD-mod-binarySwitch-v3_0_0.xsd"}]
},{
"@type": ["Action","onem2m:toggle","org.onem2m.home.moduleClass.binarySwitch.toggle"],
"name": "myToggle",
"links": [{
    "href": "myBinarySwitch/myToggle",
    "method": "PUT",
    "mediaType": "application/json",
    "schemaLocation": "oneM2M:HD-act-toggle-v3_0_0.xsd"}]
```

Alt2-flat

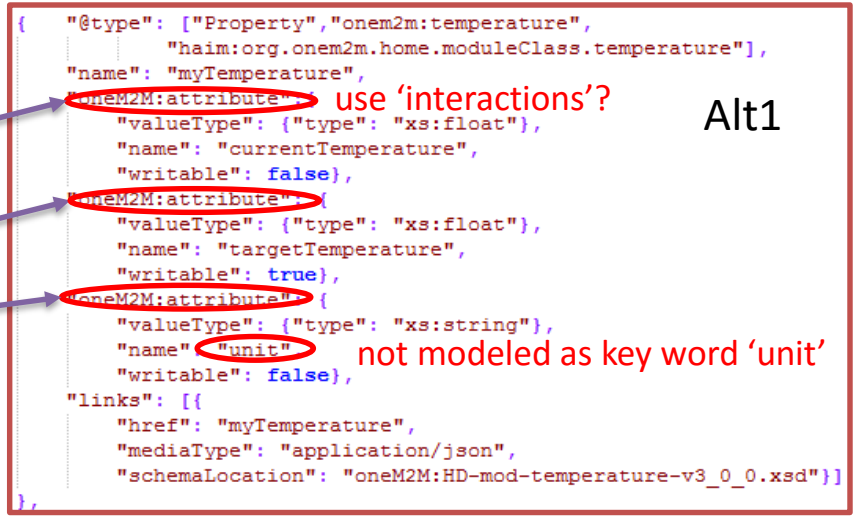
- lose resource hierarchy,
- attribute addressing not supported in oneM2M.

y mapping to WoT (4/5)

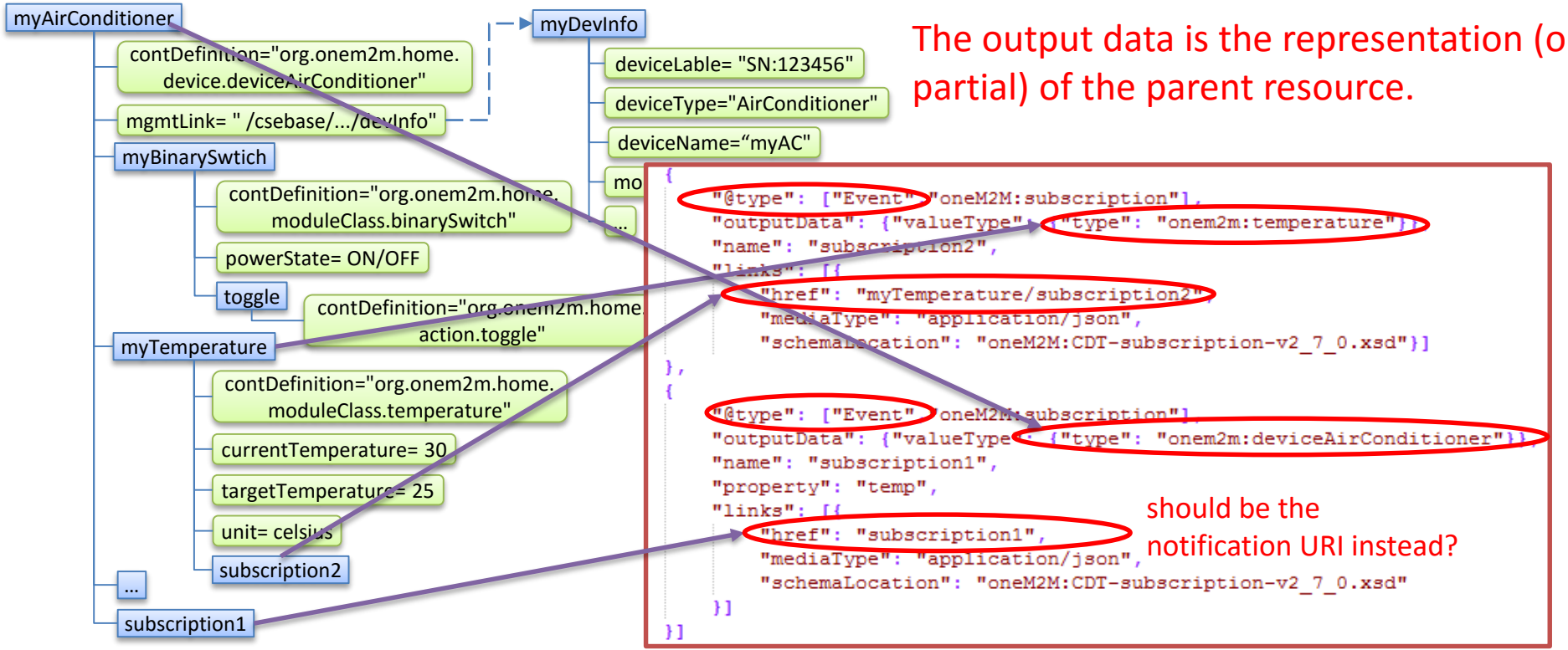




How to describe hierarchical properties, and common attributes (@type, links) ?



An exemplary mapping to WoT (5/5)



The output data is the representation (or partial) of the parent resource.

should be the notification URI instead?

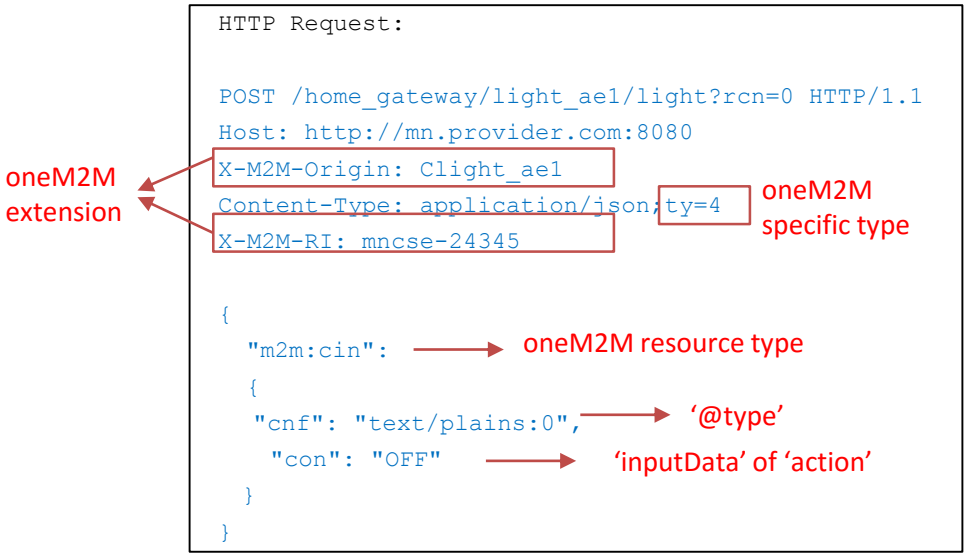
Mapping principle summary

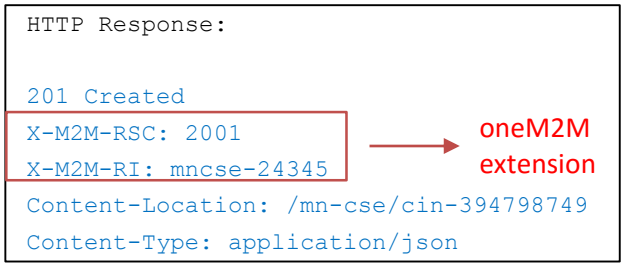
oneM2M HAIM (SDT-based)	WoT	notes
Device	Thing	There might be sub-devices in oneM2M/SDT, current TD doesn't support this.
ModuleClass	???	A collection of (structured) property/action/datapoint. Current TD seems doesn't support hierarchical representation?
Property	Property	
DataPoint	Property	
Action	Action	No clear mapping of 'outputData' in oneM2M/SDT
Event (Subscription)	Event	'Links' refer to the <subscription> resource URI or 'notificationURI'?
Resource/attribute name	Name	How to represent hierarchical resource structure of oneM2M is not clear.
Data type, resource type,	@type	
Resource uri	Links	oneM2M doesn't support attribute level addressing. TD should also provide schema info for interworking with oneM2M native i/f.

*note: mapping of ordinary oneM2M resources/attributes to TD without the modeling of HAIM is FFS.
Automated mapping may be harder due to lack of semantics.

A message example

- The following example shows a *contentInstance* resource create request and response using HTTP with JSON serialization in oneM2M.





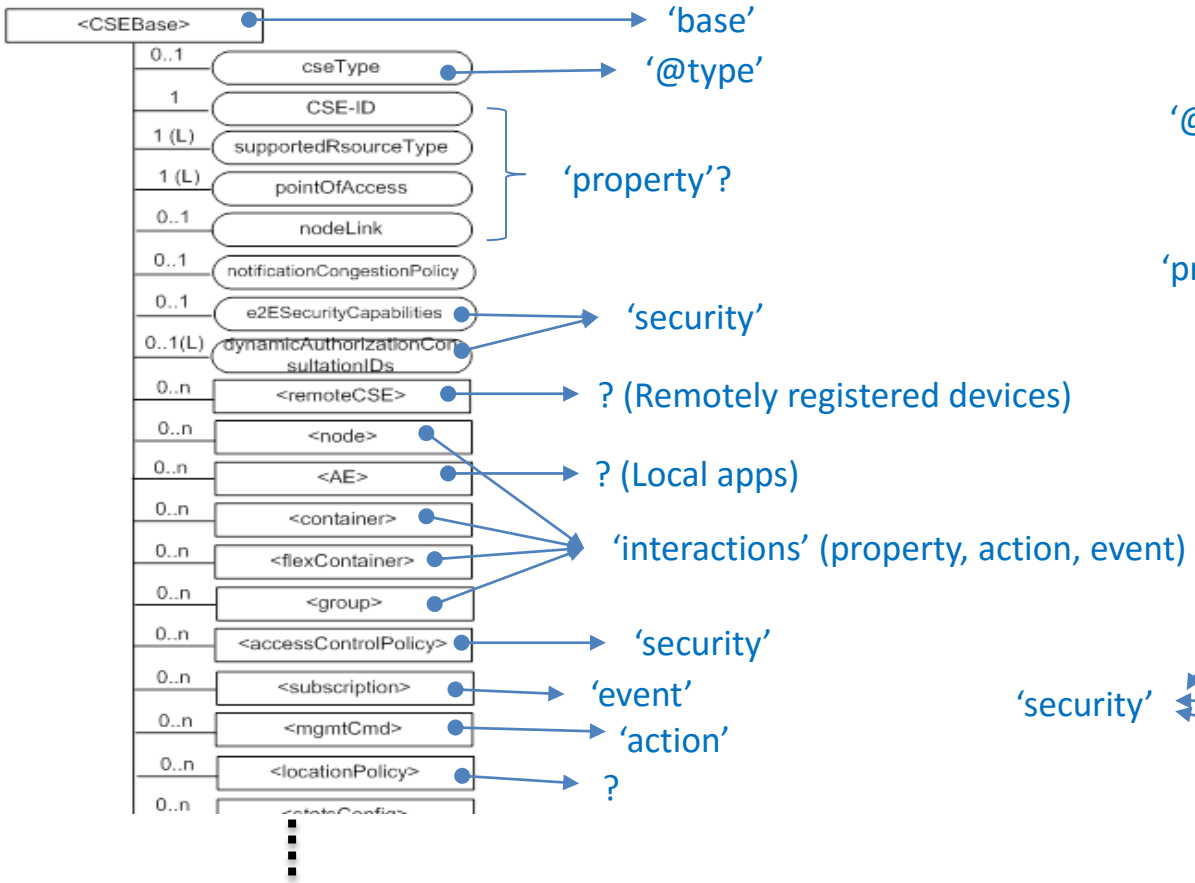
Question: How to describe platform-specific extension of protocol bindings?

Discussion

- Mapping to '@type'
  - should be able to support oneM2M specific resource types, schema, data types, semantic annotations
- Mapping to 'property'
  - oneM2M differentiates between static (non-functional) properties and dynamic (functional) data points. Is it sufficient to map everything to 'property'? Would it be beneficial to differentiate in WoT as well?
- Mapping to 'action'
  - The modeling of complex RESTful (non-RPC) action: write multiple 'inputData' and trigger 'action' at different URLs, see [firmware] update.
- Mapping to 'event'
  - whether/how to describe comprehensive event criteria and notification policies in oneM2M? What to carry in Links?
- Group vs 'associations'
- Base Ontology vs WoT Ontology
- Security
  - Whether/how to model access control policies in WoT? (not 'writable')
- Protocol bindings
  - Whether/How to describe oneM2M specific extensions to HTTP, MIME Type, …?

Thank you!

oneM2M Resource Model vs. WoT



Universal Attributes (mandatory)

'@type' → **Attribute Name**: *resourceType*

'property' →

Attribute Name
resourceID
resourceName
parentID
creationTime
lastModifiedTime
expirationTime

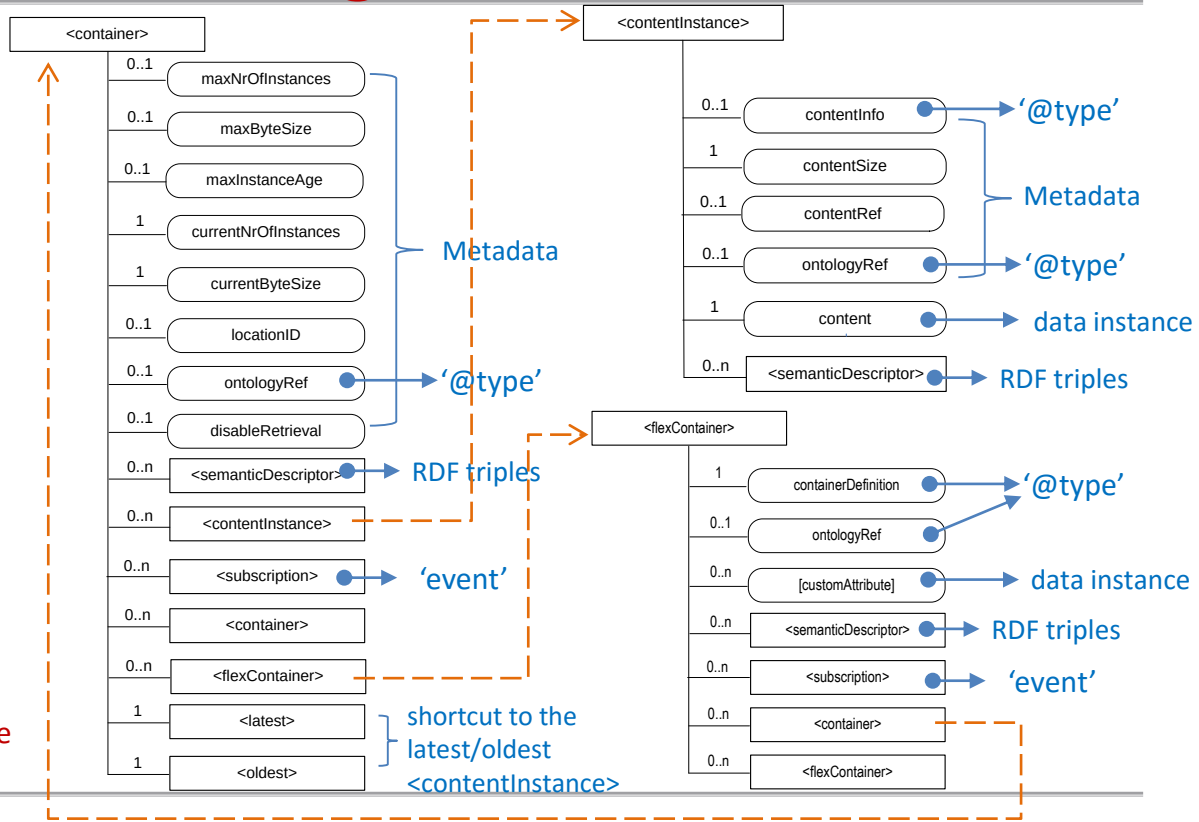
Common Attributes (optional)

Attribute Name
accessControlPolicyIDs
stateTag
announceTo
announcedAttribute
labels
e2eSecInfo
dynamicAuthorizationConsultationIDs
creator

'security' →

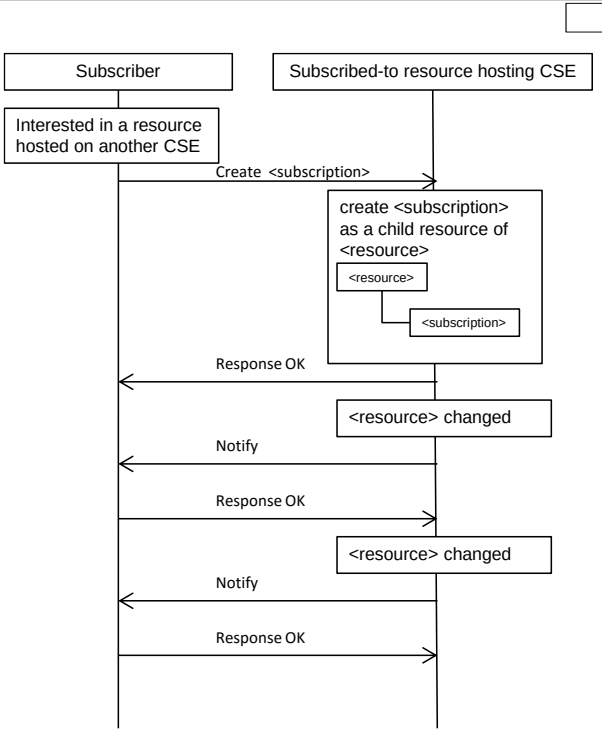
Data Management

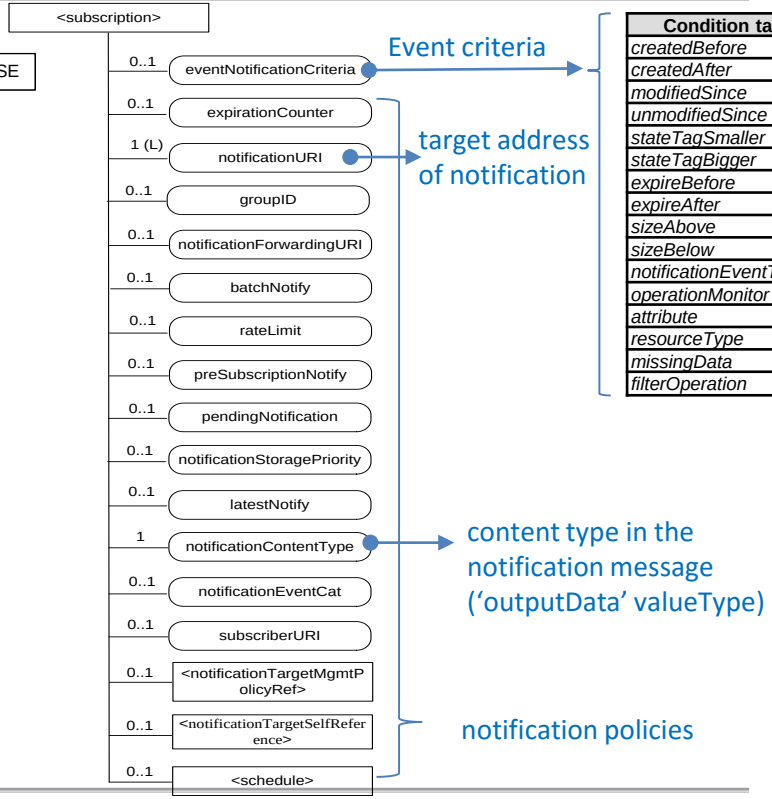
- Different resource types
  - <container> + <contentInstance>
    - multiple instances
    - rich metadata (incl. storage policy)
  - <flexContainer>
    - flatter and simpler structure
    - can be specialized to any data model
  - mixed
- Support hierarchical data model
- Support semantic annotation
- 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.

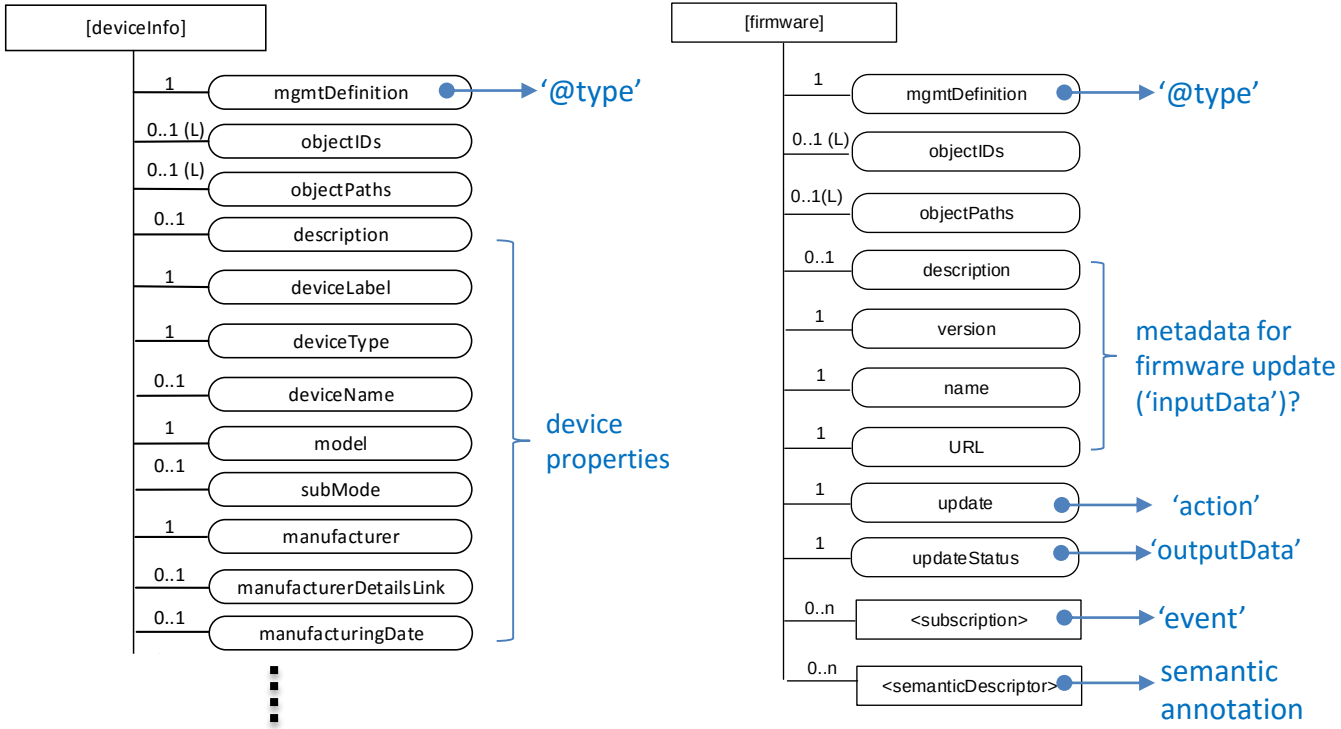




Device Management

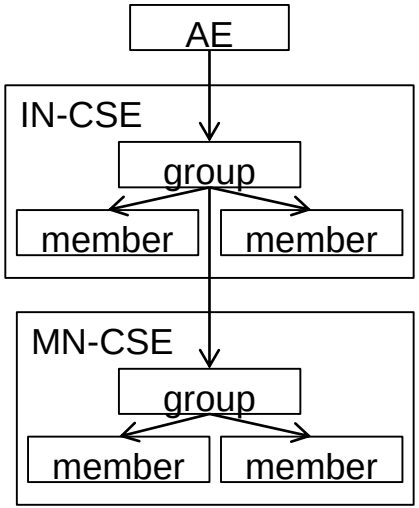
- <mgmtObj> as a template is specialized to individual management resources e.g. [deviceInfo], [firmware]
- Some are actionable, some are not.

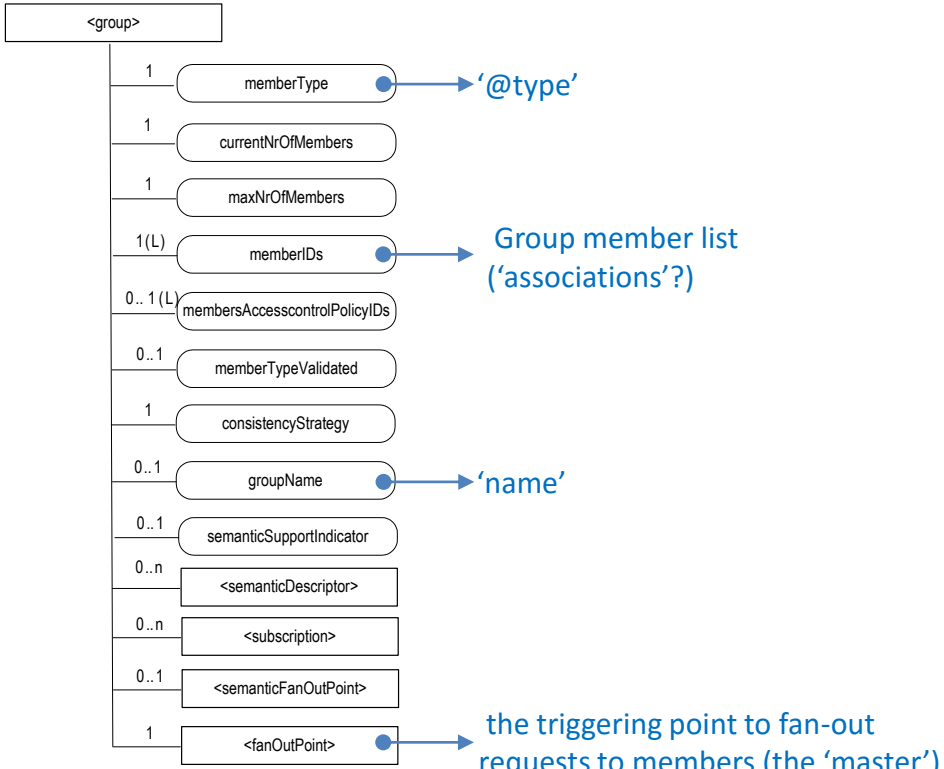
D.2 Resource firmware
D.3 Resource software
D.4 Resource memory
D.5 Resource areaNwkInfo
D.6 Resource areaNwkDeviceInfo
D.7 Resource battery
D.8 Resource deviceInfo
D.9 Resource deviceCapability
D.10 Resource reboot
D.11 Resource eventLog
D.12 Resource cmdhPolicy



Group Management

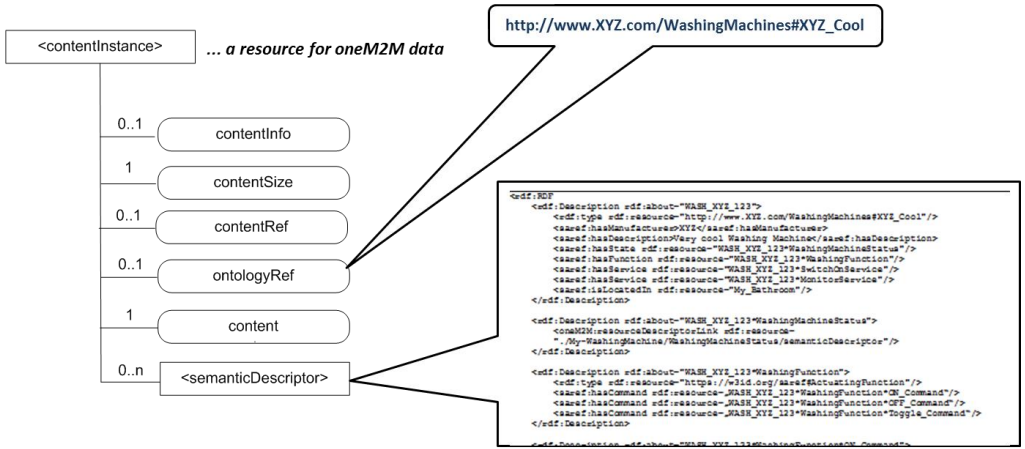
- Distribute requests to and converge responses from multiple devices via a group hosting CSE (device/gateway/platform) to improve communication efficiency





Semantics

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

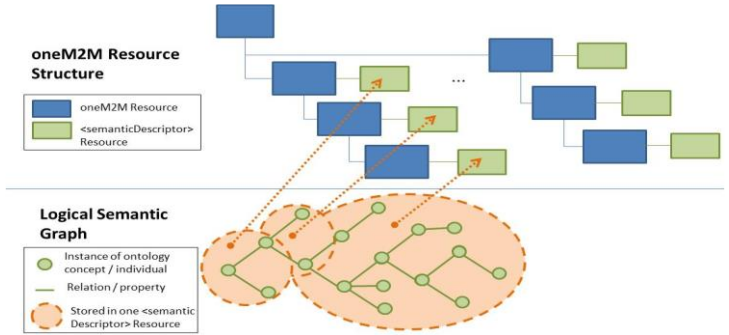


    - ✓ *A description of the data itself and its relation to other data*

… 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 …**