oneM2M Answers to WISHI State of the Union Questions

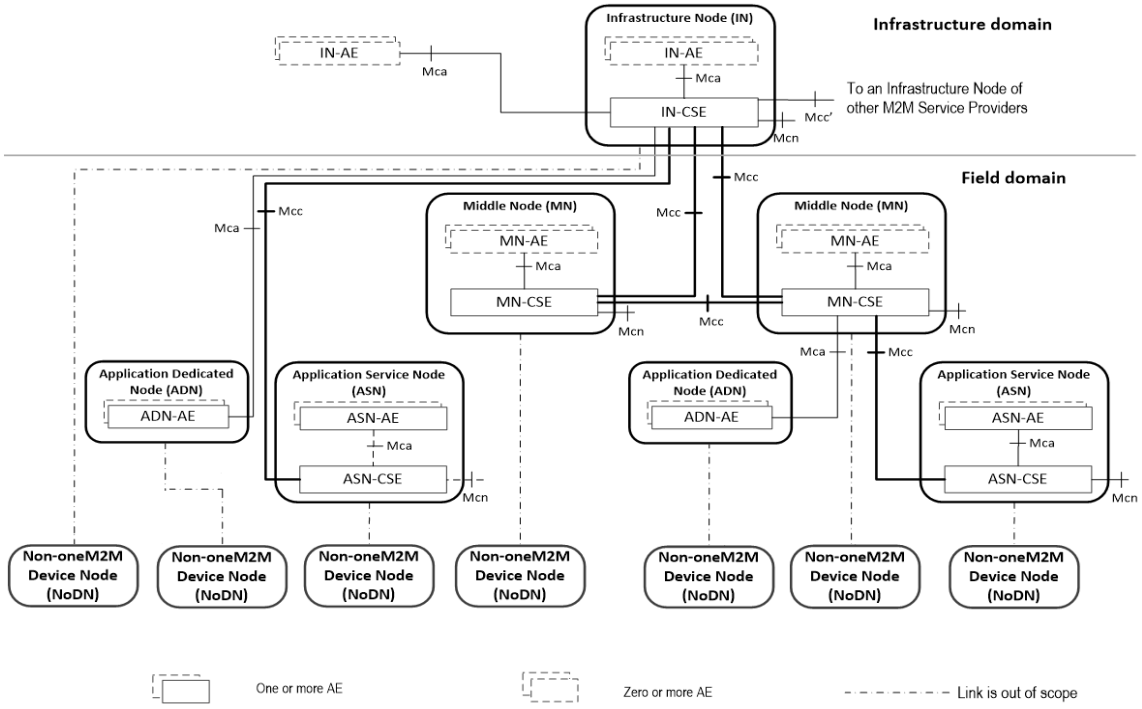
Group Name: IRTF
Source: Tim Carey, oneM2M MAS Vice-chair, timothy.carey@nokia.com
Meeting Date: July, 2017

What do you work on?

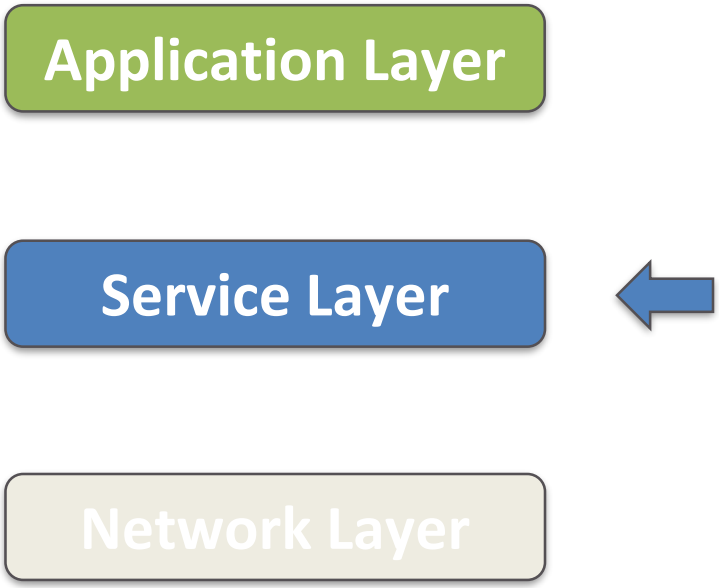
The oneM2M set of specifications provides End to End IoT Service Layer Functions that includes interface specifications between IoT applications hosted in the cloud, service provider infrastructure and devices and gateways in the field domain.

TS-0001 Functional Architecture describes the common service functions specified by oneM2M.

Specifications are online:
http://www.onem2m.org/technical/published-documents

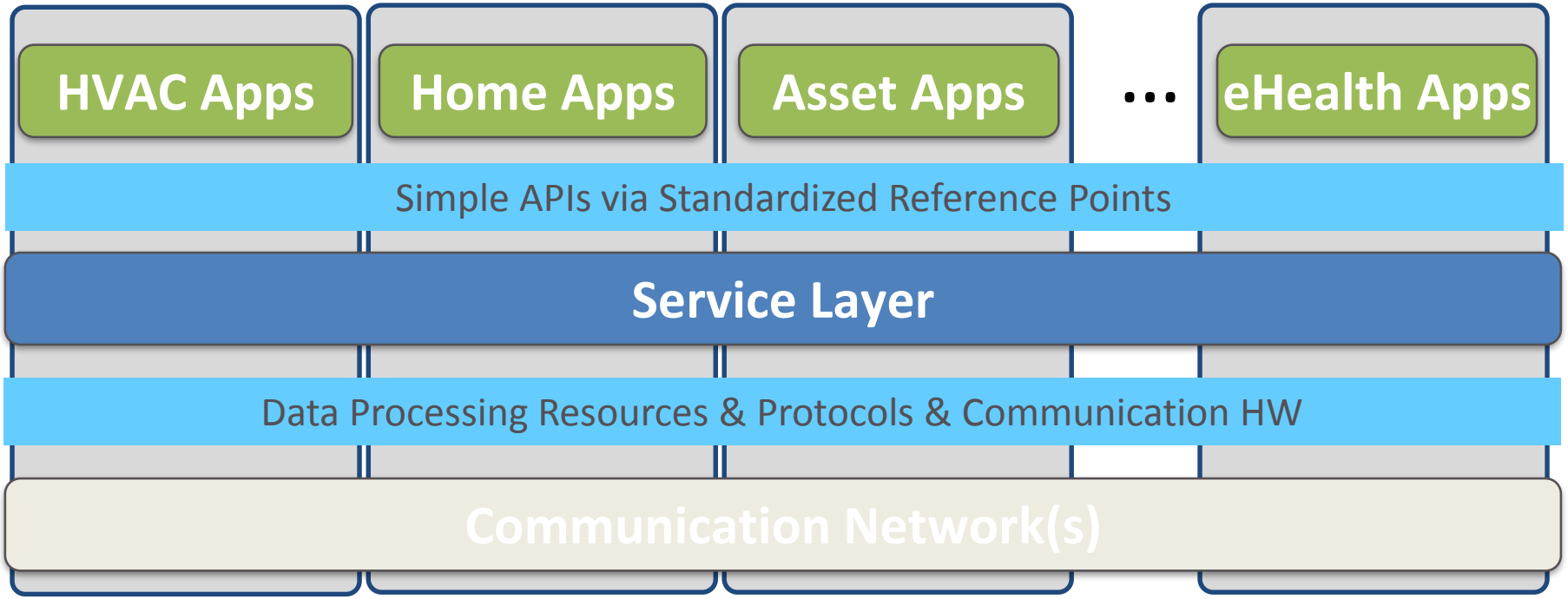


What is oneM2M?



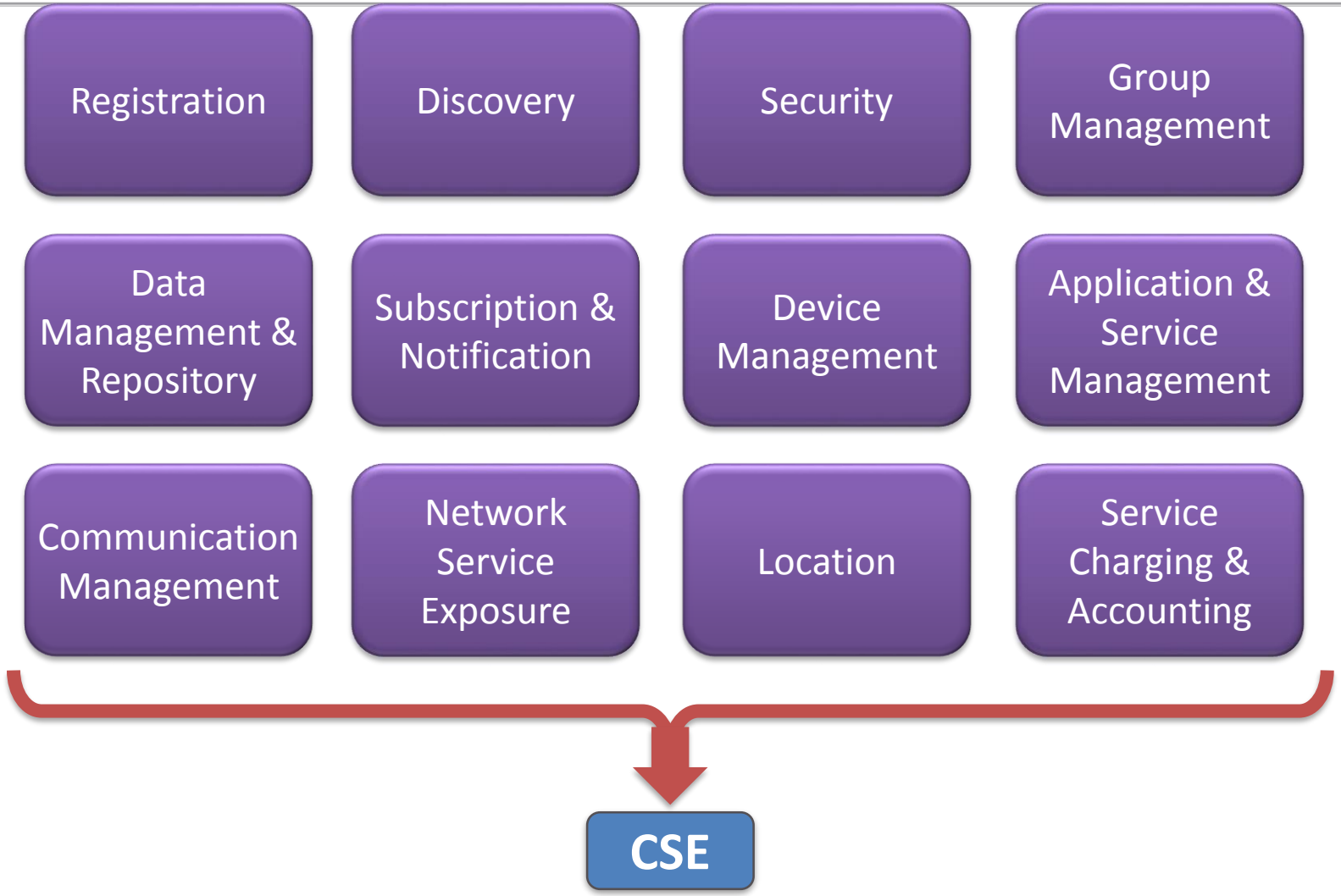
- It is a software/middleware layer
- It sits between applications and underlying communication networking HW/SW
- It typically rides on top of IP protocol stack
- It provides functions that applications across different industry segments commonly need
- It exposes common set of functions to applications via developer friendly APIs
- It is integrated into devices/gateways/servers and allows distributed intelligence
- It hides complexity of NW usage from apps
- It controls when communication happens
- It stores and shares data
- It supports access control
- It notifies applications about events

What is oneM2M?



Horizontal layer of functions commonly needed across different industry verticals

Common Service Functions



Communication Protocols

oneM2M Release 1 supports bindings to HTTP, CoAP and MQTT underlying IP-based transport protocols and XML or JSON content serializations

HTTP and JSON Example

REQUEST

```
GET   /~/CSE-178/CSEBase/home/temperature HTTP/1.1
Host: provider.net
X-M2M-Origin: /CSE-123/WeatherApp42
X-M2M-RI: 56398096
Accept: application/json
```

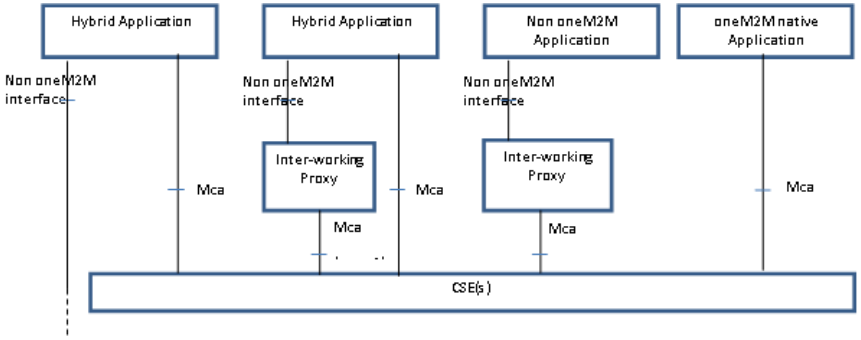
RESPONSE

```
HTTP/1.1 200 OK
X-M2M-RI: 56398096
X-M2M-RSC: 2000
Content-Type: application/vnd.onem2m-res+json
Content-Length: 101
{“m2m:cin”:[
   "cnf":"application/json:0",
  "con":"{'timestamp':1413405177000,'value':25.32}"]
}
```

Domains in the oneM2M Focus?
Horizontal Service Layer

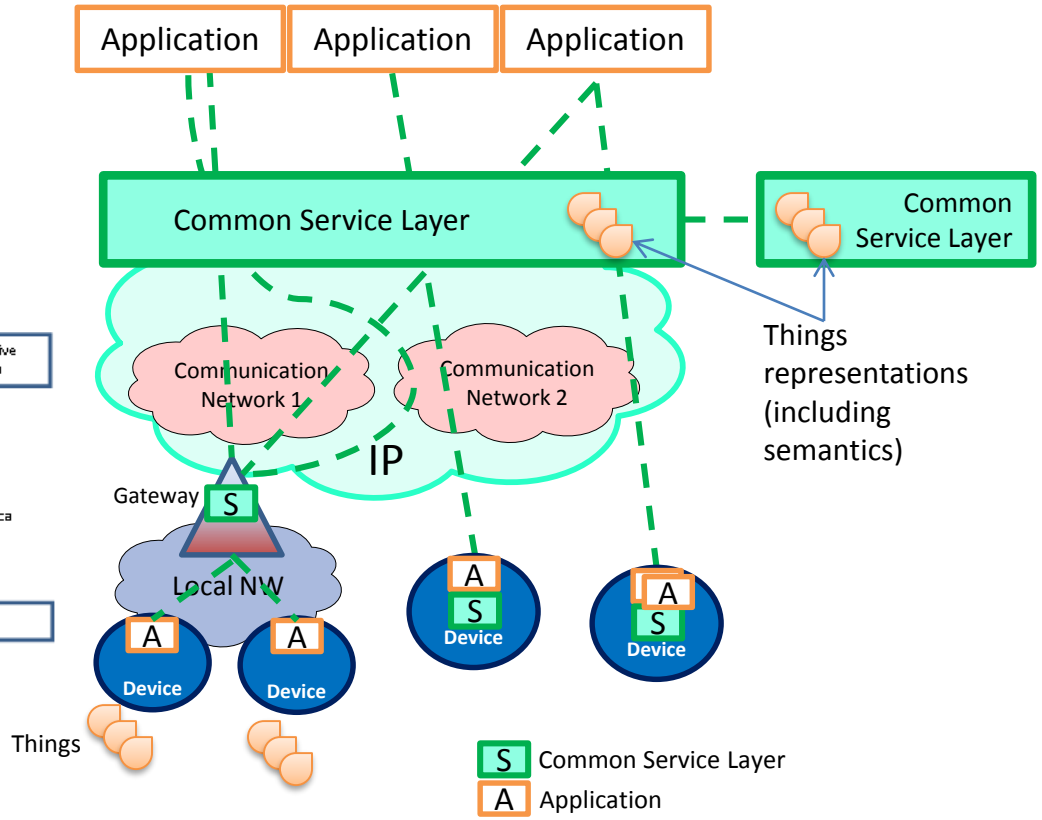
oneM2M is a Horizontal platform. By its architectural approach the specifications are applicable to most any domains that are IP based.

For non-IP Domains oneM2M provides a specifications and services for interworking with these domains.

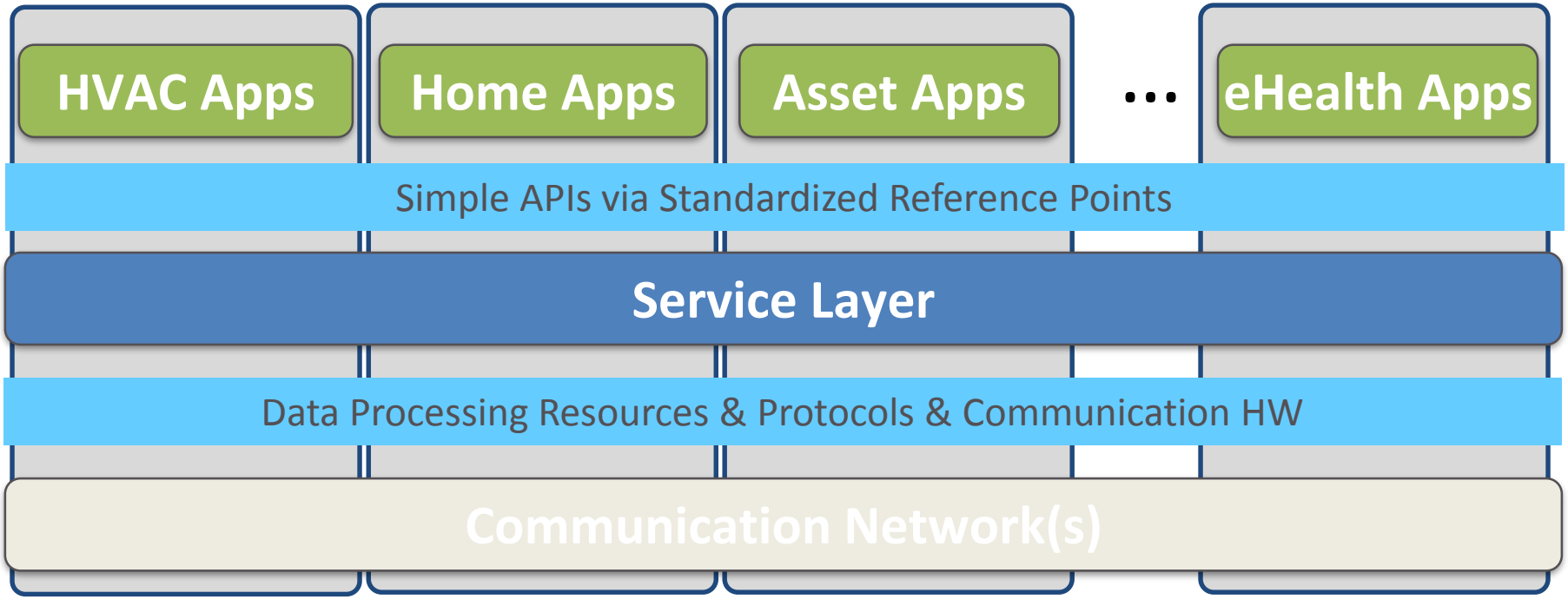


Horizontal (based on common Layer)

Applications share common service and network infrastructure
Multipoint communications

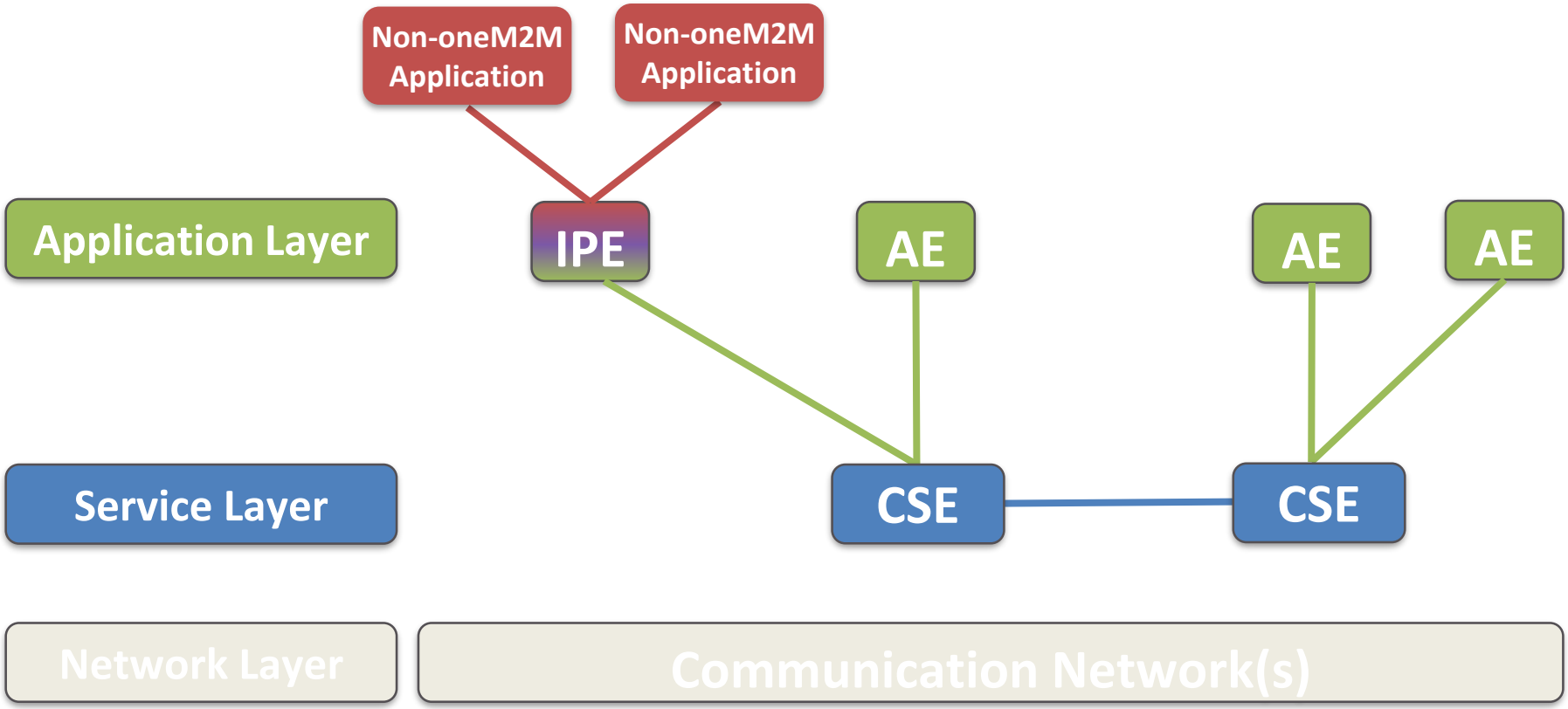


Domains in the oneM2M Focus: Horizontal Service Layer



Horizontal layer of functions commonly needed across different industry verticals

Domains in oneM2M Focus: Interworking



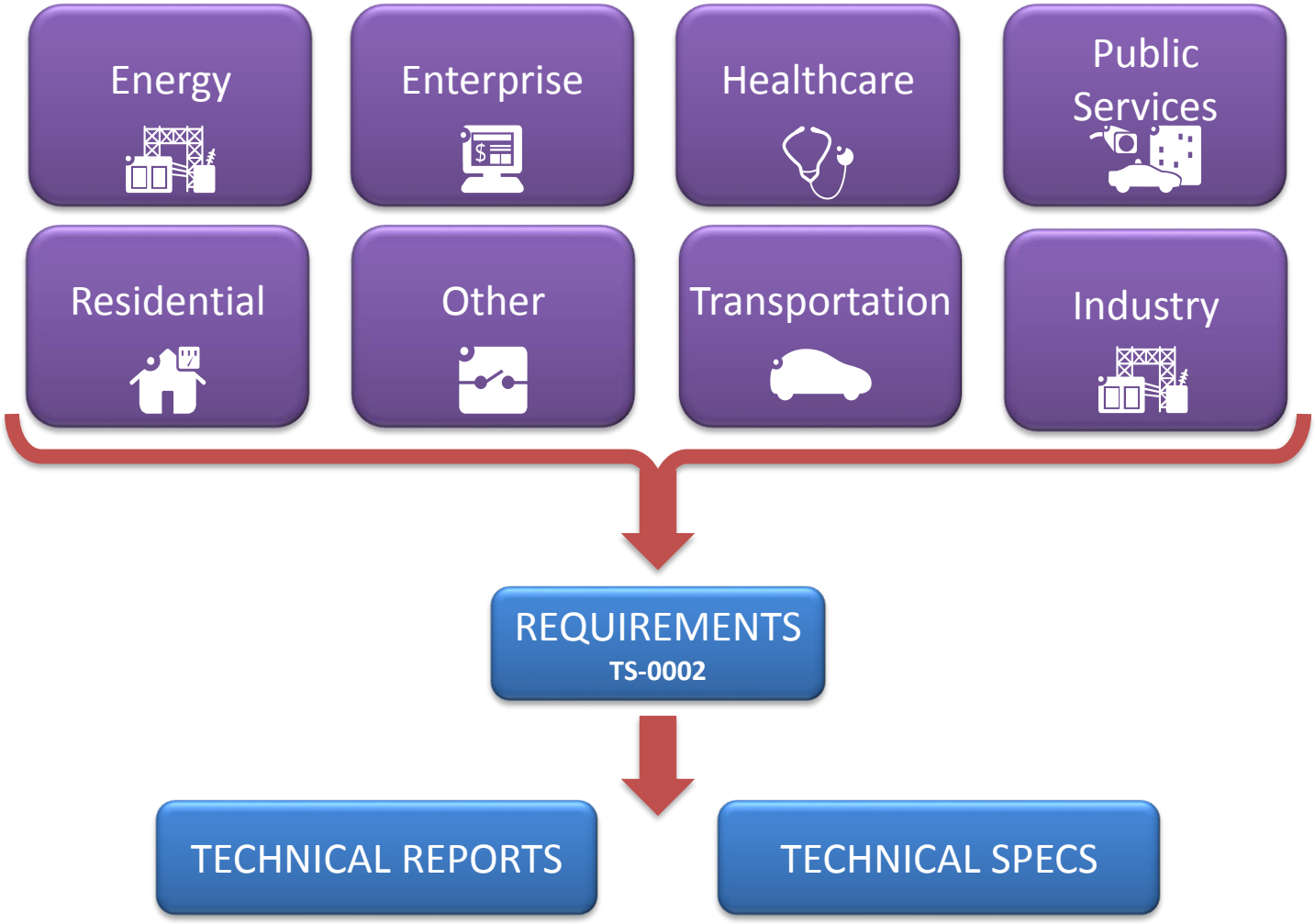
IPE: Interworking Proxy Application Entity

AE: Application Entity

CSE: Common Services Entity

Domains in the oneM2M Focus?

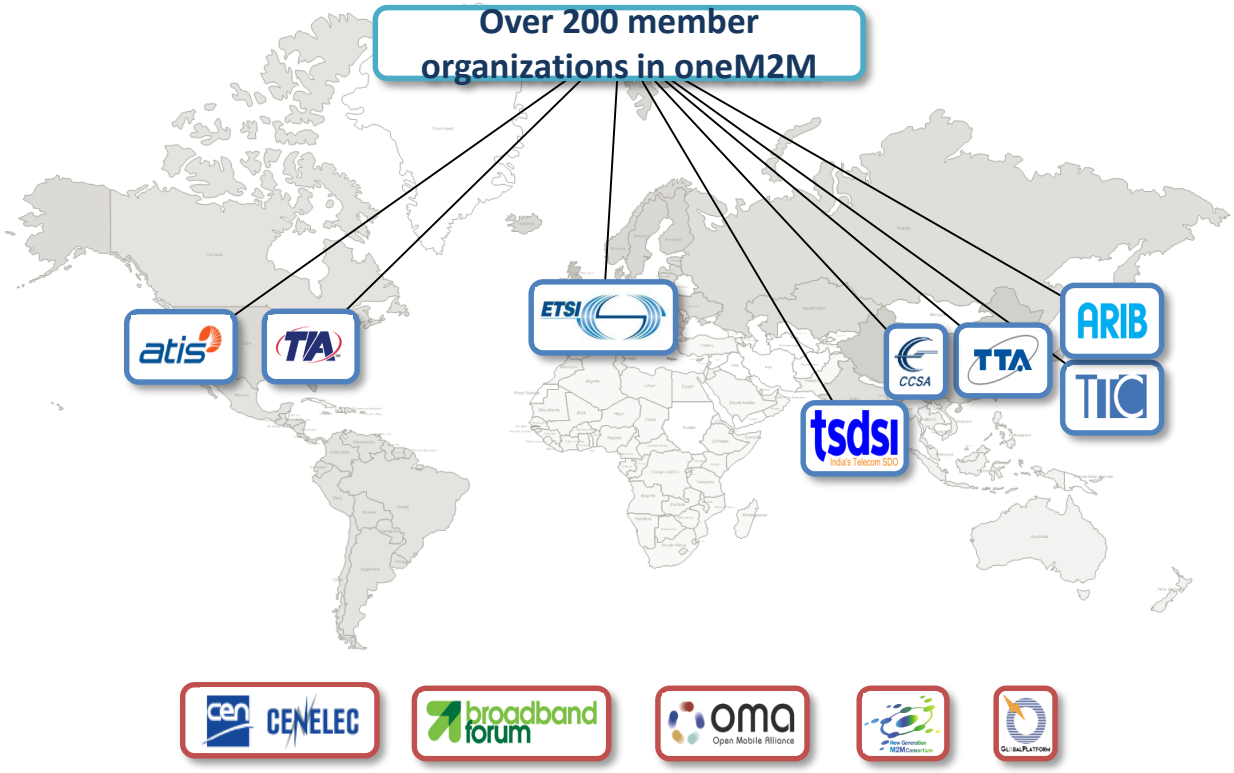
While oneM2M is a Horizontal service layer: The following Vertical Domains have been considered in development of the specifications.



All document are publically available

How do you foster interoperability?

oneM2M is a partner project that brings together global standards organizations as partners.

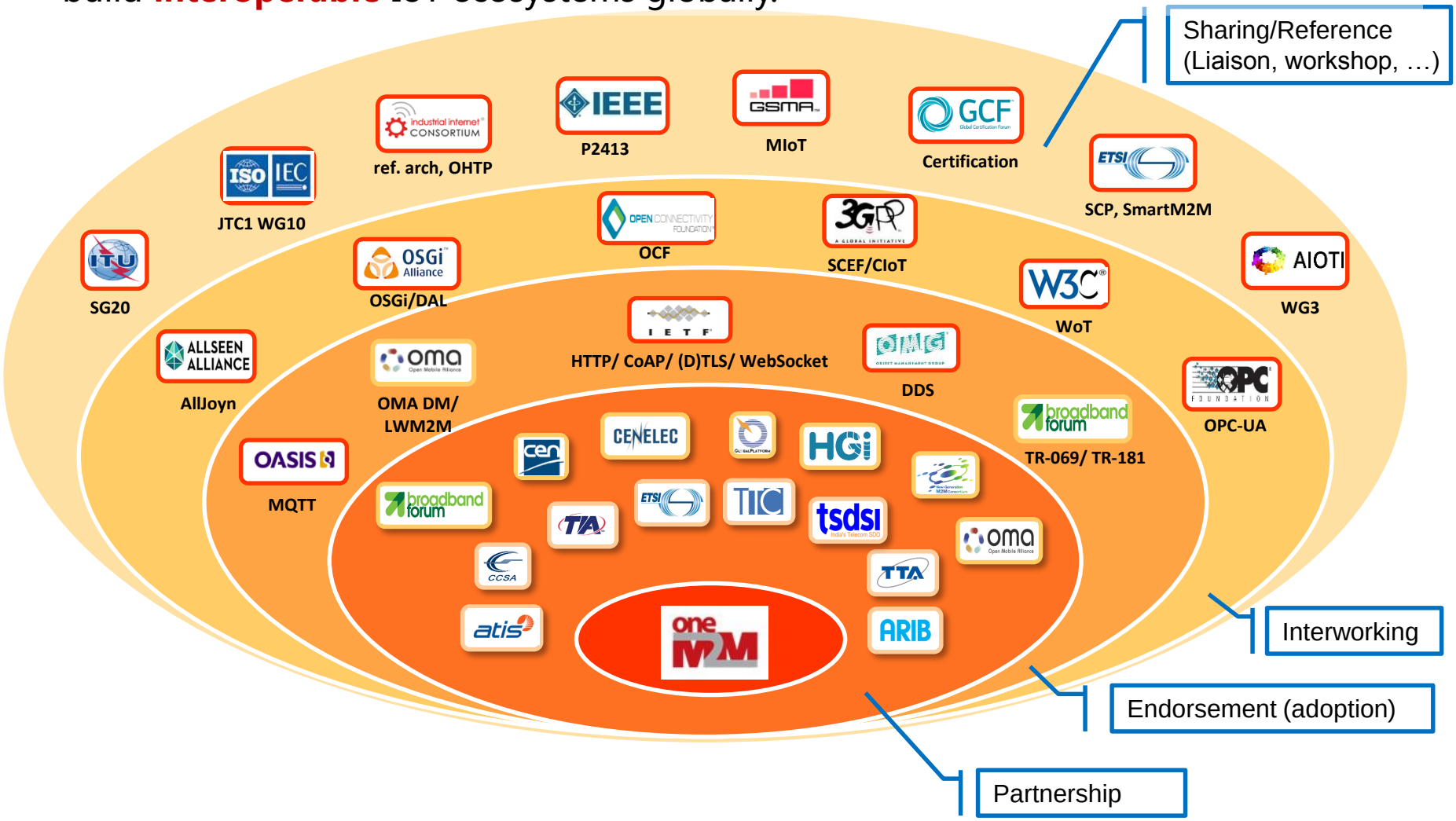


www.oneM2M.org

All document are publically available

Ongoing Collaborations

- Collaboration is important to reach common understanding, avoid overlap and build **interoperable** IoT ecosystems globally.



How do you foster interoperability? Strong opens source implementation base

Industry-driven Open source implementations

LAAS-CNRS

Fraunhofer FOKUS

KETI

CISCO

OM2M Connecting things

open mtc

OCEAN

OPEN DAYLIGHT

IotDM

Examples of Commercial implementations /demos

LG

सेंटर फॉर डेवलपमेंट ऑफ टेलीमैटिक्स
भारत सरकार का दूरसंचार प्रौद्योगिकी केन्द्र
सी-डॉट C-DOT
Centre for Development of Telematics
Telecom Technology Centre of Govt. Of India

PILOT THINGS

SIERRA WIRELESS

InterDigital

QUALCOMM

kt

HUAWEI

SK telecom

sensinov
Global IoT Platform

MODACOM

LG U+

Hewlett Packard Enterprise

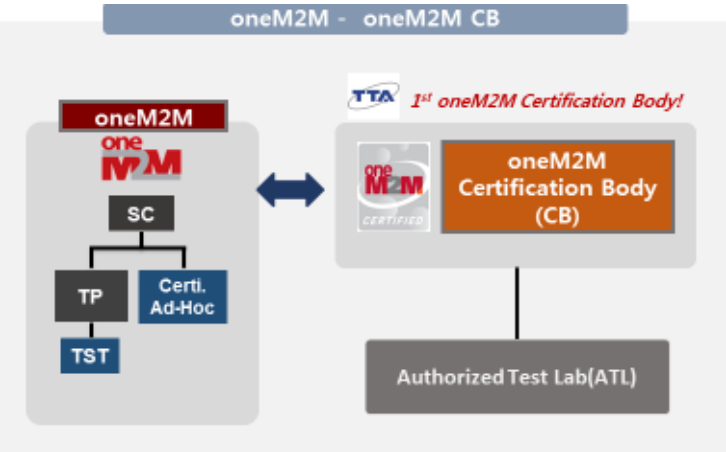
5 interop. events so far

How do you foster interoperability? Formal Certifications

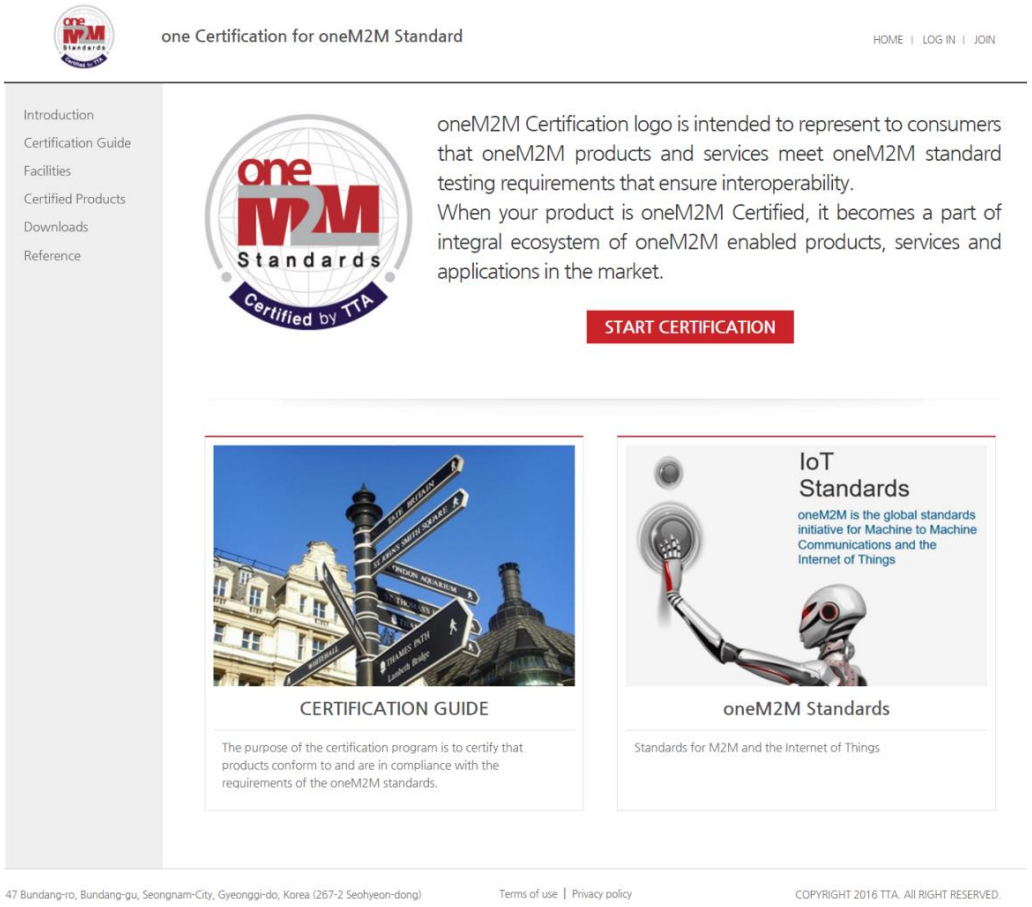
- oneM2M Certification logo is intended to represent to consumers that oneM2M products and services meet oneM2M standard testing requirements that ensure interoperability.

How do you foster interoperability? Formal Certifications

- 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 (e.g. GCF) to be setup in 2018.
- See oneM2M certified products at: www.oneM2Mcert.com



www.oneM2Mcert.com



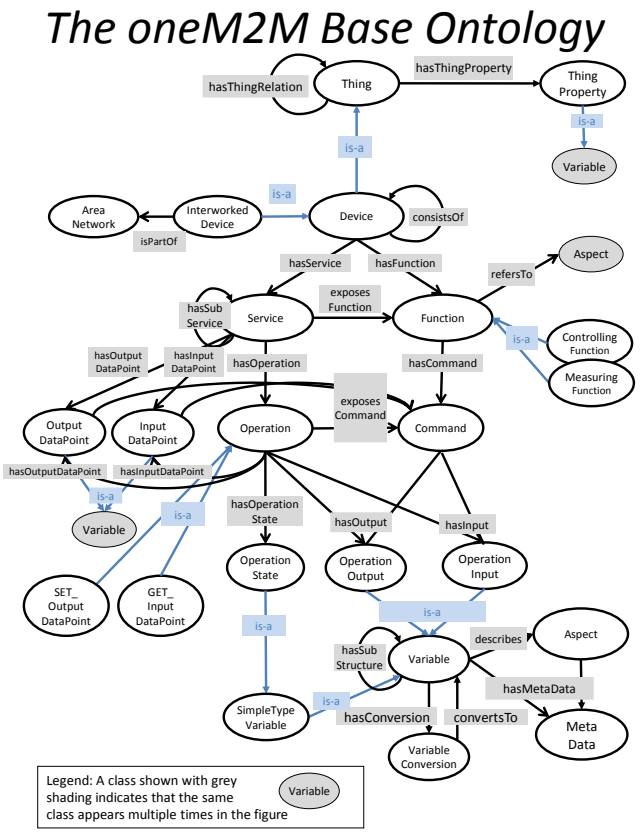
How do you foster interoperability? Registration of Applications

- oneM2M protocols require unique identification of oneM2M applications. The use of non-standard identifiers and proprietary formats for identifying software applications makes interconnection extremely difficult, as well as not necessarily being unique. It also prevents effective tracking and reporting necessary for service fulfilment and billing.
- The oneM2M Application-ID Registry solves the problem by providing a central source for application registrations and subsequent lookups. The registry enables:
  - Generation of unique standards-based identifiers
  - Centralized App-ID data management through a robust, fault-tolerant registry
  - Processing of thousands of concurrent transactions
- ATIS, a oneM2M founding partner, is the initial Registry Management Authority for the oneM2M App-ID Registry powered by iconectiv.

Interoperability Topics for the IRTF

- oneM2M provides services for interoperability between Applications using:
  - Development of Information models for specific domains (e.g, Home Automation, V2X, Industrial domain) allows Applications to interact with devices using the same representation of the resource. Regardless of the type of device (e.g., oneM2M, Zigbee, Alljoyn)
  - Automated generation of resources using the Smart Device Template: Information models are defined using the Smart Device Template which then generates the software artifacts needed to express the resource in the oneM2M system.
  - Semantics: Through semantic annotation, using the oneM2M ontology, of the resources represented by the information model; semantic meaning can be discovered, queried and mashed-up.

Work on Semantics - Ontologies

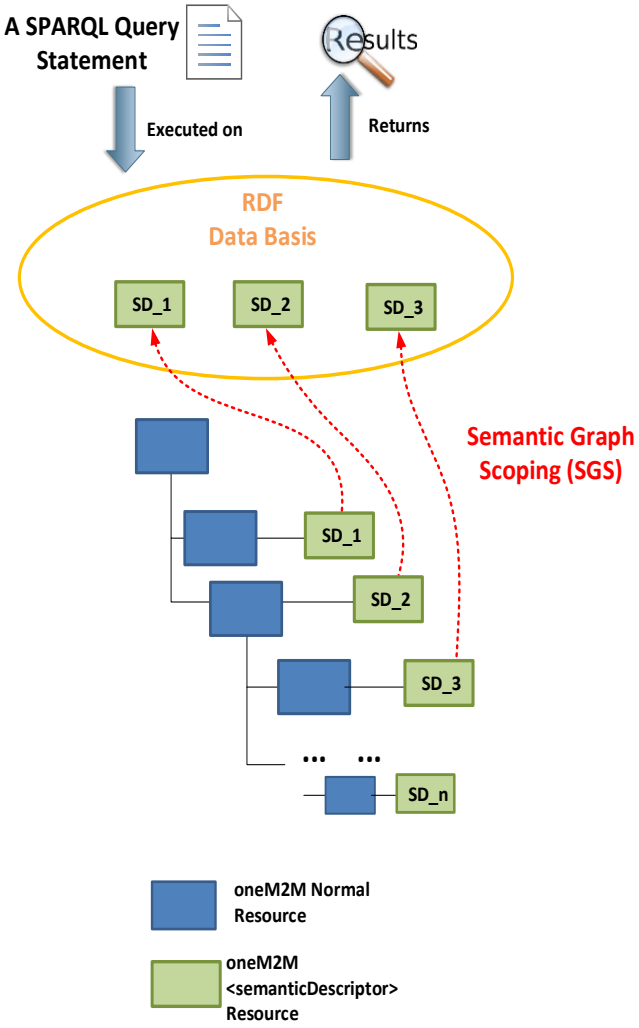


- oneM2M allows to ***annotate*** application specific resources (M2M data) with semantic description.
  - Uses a specialized resource type *<semanticDescriptor>*
  - Can contain proprietary semantics

  or
  - Semantics according to a published ontology
- The oneM2M ***base ontology*** is a top-level ontology that allows to create sub-classes (or equivalence classes) for application-level ontologies
  - Example: Smart Appliances Reference Ontology (SAREF)
- Ontologies can be used in oneM2M to describe the application specific data model of an external system for the purpose of interworking.
  - oneM2M ***Generic Interworking*** uses such an ontology to enable interworking of oneM2M entities with devices of the external system

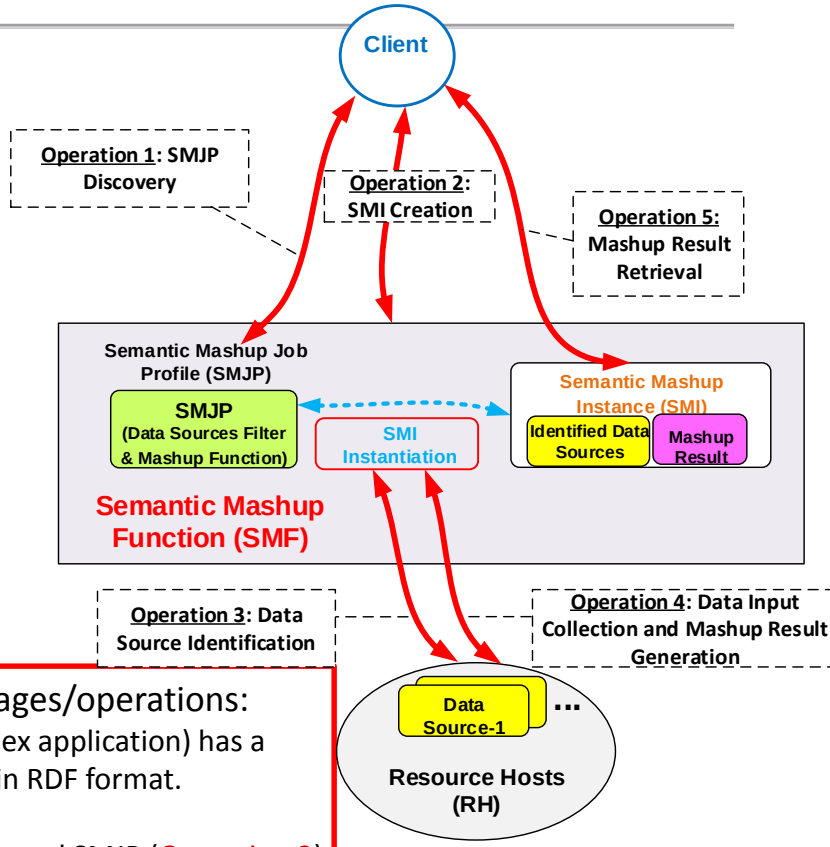
Work on Semantics Query

- oneM2M includes a semantic query feature that includes both discovery and query capabilities
  - Semantic resource discovery is used to discover resources: Give me the resources that represent the temperature sensors located in Room 1.
  - Semantic query is used to extract "useful knowledge" (to answer the query) over a set of "RDF data basis". What is the manufacture name and production year of the temperature sensors located in Room 1?
- To successfully execute a semantic query requires appropriate semantic graph scoping and extra information represented in RDF triples
  - Semantic Graph Scoping: How to collect RDF triples from semantic descriptors (distributed in the resource tree) to construct a RDF data basis for a given semantic query.
  - Representing Extra Information in RDF Triples: This is for how to query information that was originally not stored as RDF triples, such as data stored in *<contentInstance>* resource (or other oneM2M attributes such as *expirationTime*, etc.).



Work on Semantics Mashup

- oneM2M supports semantic mashup, which fully leverages semantic-related technologies and is defined as a process to discover and collect data from more than one source as inputs, conduct business logic-related mashup function over the collected data, and eventually generate meaningful mashup results.
- Example: Users/clients are interested in a metric called "weather comfortability index", which cannot be directly provided by physical sensors, and in fact can be calculated based on the original sensory data collected from multiple types of sensors (e.g. temperature and humility sensors).



- A complete semantic mashup process consists of multiple stages/operations:
  - Each specific mashup application (e.g. weather comfortability index application) has a corresponding Semantic Mashup Job Profile (SMJP) represented in RDF format.
  - A client first discovers a SMJP based on her needs (Operation 1).
  - A Semantic Mashup Instance (SMI) is created based on the discovered SMJP (Operation 2).
  - The data sources are identified for the created SMI (e.g. through semantic resource discovery) based on the guideline as specified in the SMJP (Operation 3).
  - The data is collected from the identified data sources and mashup result is generated and may be periodically refreshed based on the mashup function as specified in the SMJP (Operation 4). The mashup result can be retrieved by the client (Operation 5).

Work on Semantics Access Control

- In oneM2M, access to resources needs to follow Access Control Policies (as defined in ACP resources).
- An unique issue in oneM2M for semantic query is that for a given semantic query, ACPs still need to be enforced in the sense that only the RDF triples stored in certain <*semanticDescriptor*> resources (i.e., allowed by ACP) can serve as the RDF data basis for this query.
- In oneM2M, two solutions are identified to solve this ACP-related issue when processing a semantic query:
  - 1. ACPs are still kept in their original form and stored in the resource tree. For each received semantic query, the Hosting CSE directly decides which <*semanticDescriptor*> resources are allowed by the ACP to form the RDF data basis for this query.
    - **Operations at the Hosting CSE:** Receive a semantic query -> semantic graph scoping with ACP (i.e. search <*semanticDescriptor*>*s* which are allowed by ACP) -> Generate RDF basis -> Execute the semantic query over RDF basis.
  - 2. ACPs stored in the resource tree are cloned and re-represented as RDF triples and directly stored in the Triple Store so that the semantic query with access control can be fully processed in the Triple Store.
    - **Operations at the Hosting CSE:** Clone ACP in RDF triples and store them in the Triple Store -> Receive a semantic query -> Modify the semantic query by adding ACP constraints in the query statement -> Execute the modified semantic query over all RDF triples in the Triple Store.

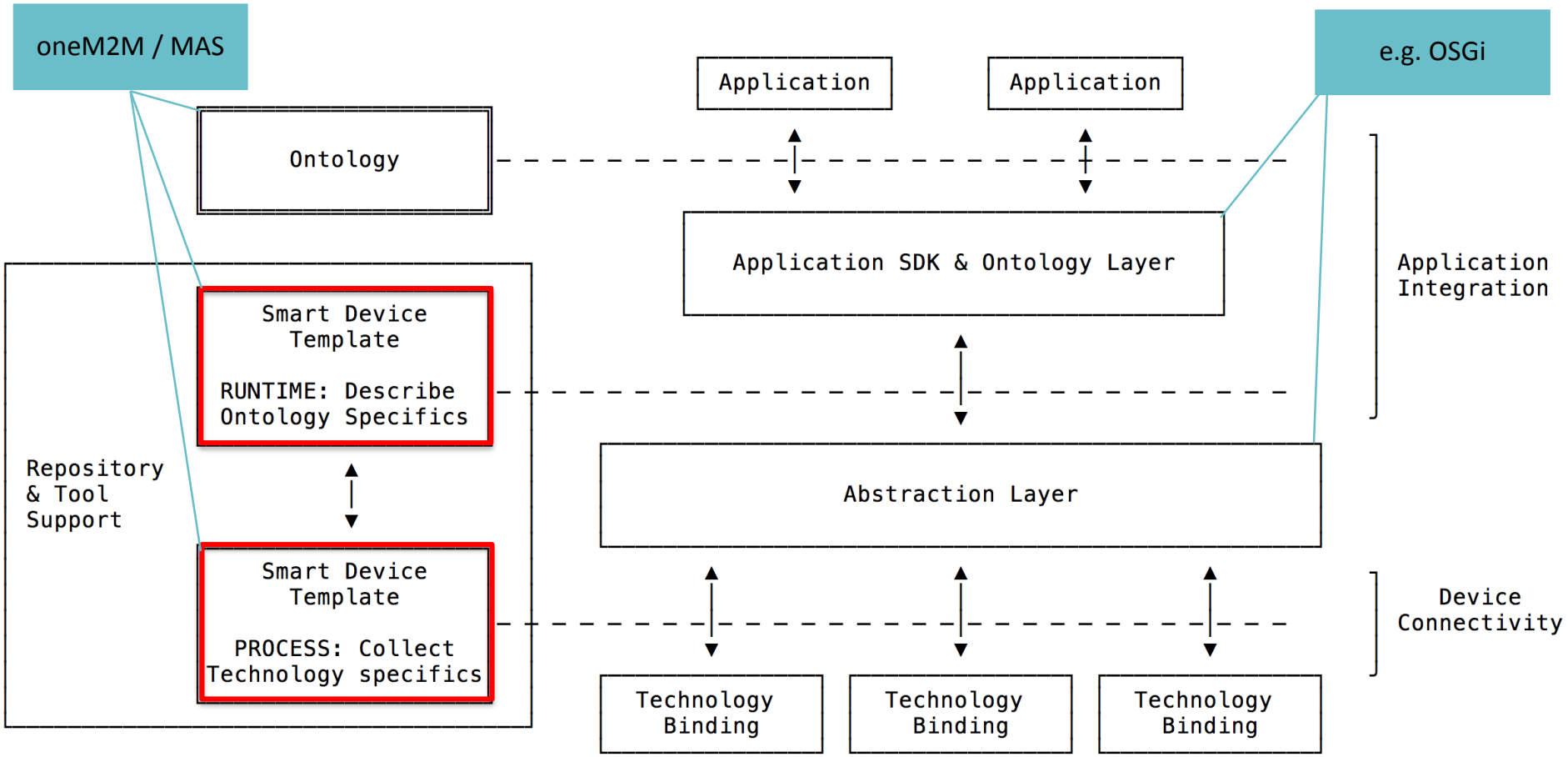
Work on Abstraction using SDT:Goals

Describe devices and device services in a way which is independent of the LAN technology in a format which is convenient and reliable for integration.

1. **Keep it simple**, especially for manufacturers to contribute
2. **Modularity** for functions and device types
3. Make it easy for developers to **create** unified APIs
4. Be **independent** of underlying home-area network technologies

Make it available under an **open license**.

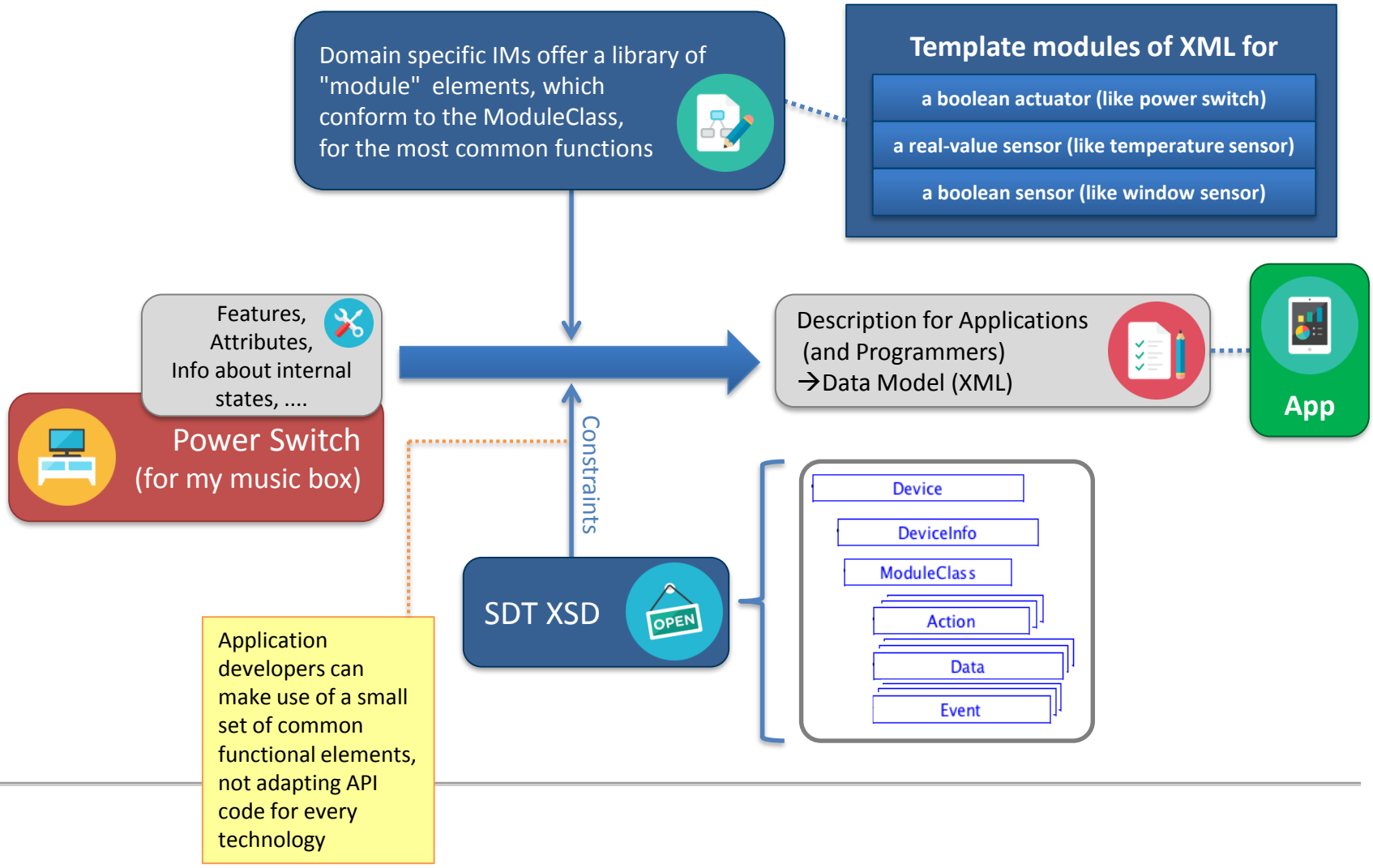
How Things Come Together



SDT: Basic Components

Domain	Unique name, or "wrapper" which acts like a namespace, set by the organization creating the SDT, allowing reference to a package of definitions for the contained ModuleClasses and device definitions. Can be referenced when extending ModuleClasses. It has two possible uses: to select the scope of a technology domain, or to set the scope of a use case domain (like Home, SmartGrid, etc)
Device & Sub-Device	Physical, addressable, identifiable appliance/sensor/actuator, that has one or more functionalities.
ModuleClass	Specification of a single service with one or more service methods, the involved abstracted data model and related events. The expectation is that each separate service which may be used in many kinds of Devices (like PowerON/OFF, Open/Close, ...) will be described by a ModuleClass which can be re-used in many Device definitions.
DataPoint	A DataPoint element represents an aspect of a device which can be read/written to, and forms part of a device's data model. Manipulating DataPoints is the most common way of controlling devices. Each DataPoint has an associated type which facilitates data integrity.
Action	Action elements are an efficient way of describing arbitrary sequences of operations/methods; these are very common in automation. Actions preserve transaction integrity by putting together all the parameters ("args", see next section) with the method which checks and executes them, in one step.
Property	Property elements are used to append to Devices and their ModuleClass elements with arbitrary additional information.

Re-usable XML Modules

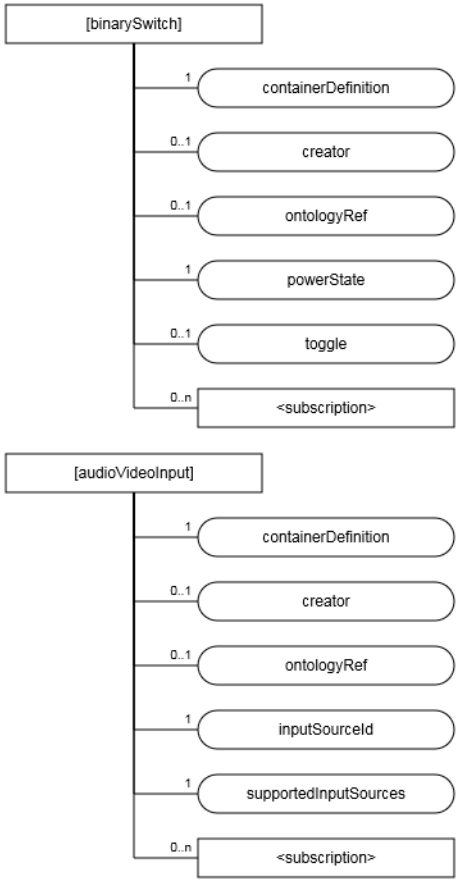


Information Modelling in oneM2M

SDT Modeling

- **Property**
  - country
  - deviceID
  - deviceType
  - deviceName
  - deviceModelName
  - …
- **Module**
  - binarySwitch
  - audioVolume
  - televisionChannel
  - audioVideoInput
  - mediaSourceList

Resource Mapping



SDT Mapping

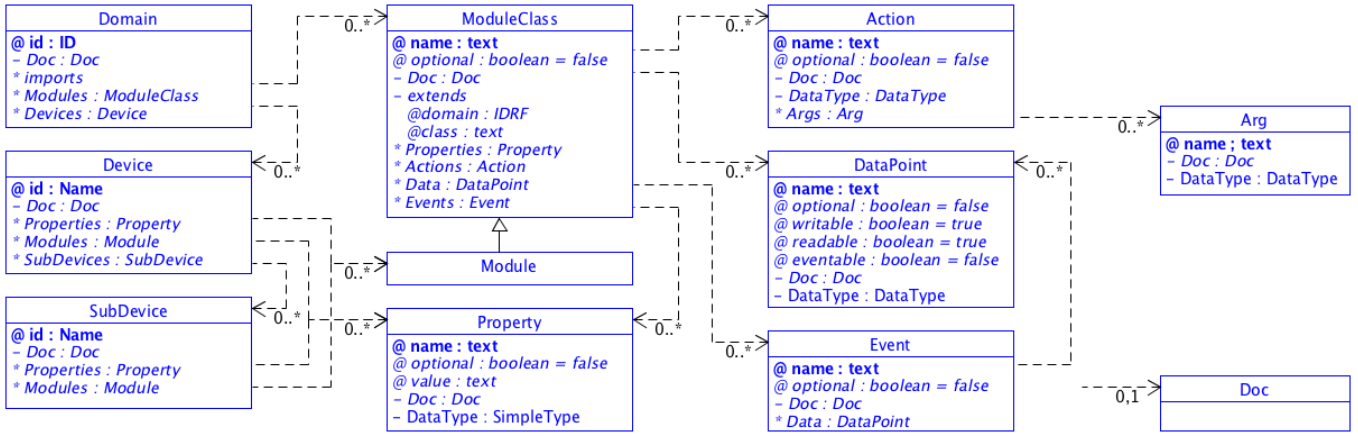
```
<ModuleClass name="binarySwitch">
    <Doc>This ModuleClass provides capabilities to control and monitor the state of
    power.</Doc>
    <Actions>
        <Action name="toggle" optional="true">
            <Doc>Toggle the switch.</Doc>
        </Action>
    </Actions>
    <Data>
        <DataPoint name="powerState" readable="true" writable="true" eventable="true"
        optional="false">
            <Doc>The current status of the binarySwitch. "True" indicates turned-on,
            and "False" indicates turned-off.</Doc>
            <DataType>
                <SimpleType type="boolean" />
            </DataType>
        </DataPoint>
    </Data>
</ModuleClass>
```

XSD Mapping

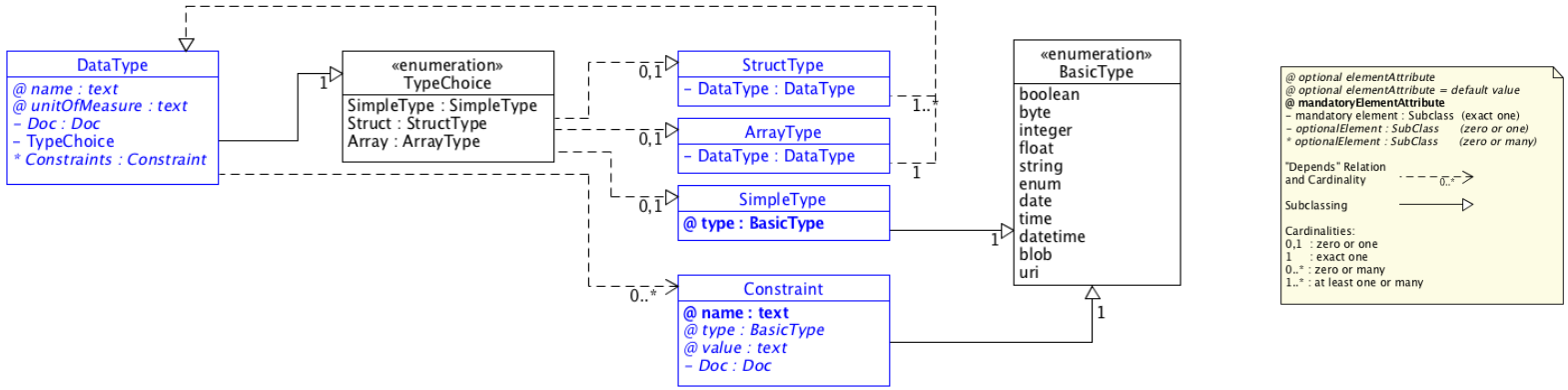
```
<xs:element name="binarySwitch" type="hd:binarySwitch" />
<xs:complexType name="binarySwitch">
    <xs:complexContent>
        <xs:extension base="m2m:flexContainerResource">
            <xs:sequence>
                <!-- Resource Specific Attributes -->
                <xs:element name="powerState" type="xs:boolean" />
                <!-- Child Resources -->
                <xs:choice minOccurs="0" maxOccurs="1">
                    <xs:element name="childResource" type="m2m:childResourceRef" minOccurs='
                    1" maxOccurs="unbounded" />
                    <xs:choice minOccurs="1" maxOccurs="unbounded">
                            <xs:element ref="hd:toggle" />
                        <xs:element ref="m2m:subscription"  />
                    </xs:choice>
                </xs:choice>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

SDT as UML

Components

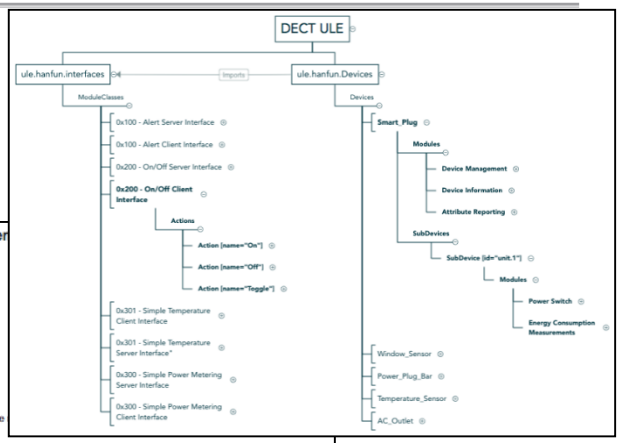


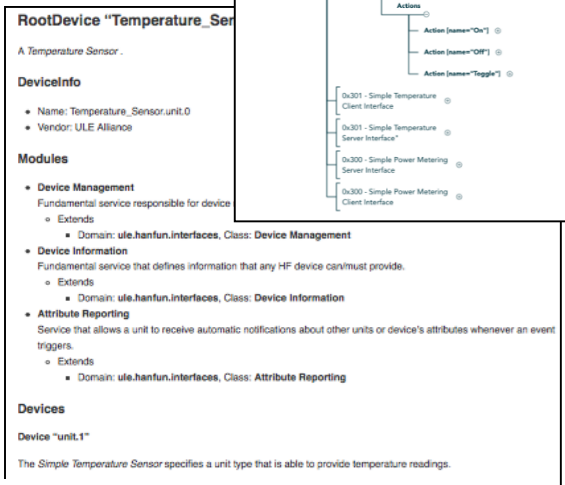
Data Types

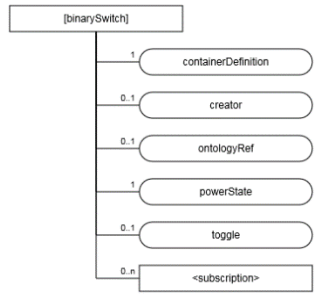


SDT Tool

- Generate oneM2M XSD
- Generate documentation
  - Plain Text
  - Markdown
  - OPML (Mindmap)
  - SVG for oneM2M Resources
- Generate code templates
  - Java interfaces and classes
  - Swagger








```
<xs:element name="binarySwitch" type="hd:binarySwitch" />
<xs:complexType name="binarySwitch">
    <xs:complexContent>
        <!-- Inherit Common Attributes from data type "flexContainerResource" -->
        <xs:extension base="m2m:flexContainerResource">
            <xs:sequence>

                <!-- Resource Specific Attributes -->

                    <xs:element name="powerState" type="xs:boolean" />

                <!-- Child Resources -->

                <xs:choice minOccurs="0" maxOccurs="1">
                    <xs:element name="childResource" type="m2m:childResourceRef" minOccurs="1" maxOccurs="unbounded" />
                    <xs:choice minOccurs="1" maxOccurs="unbounded">

                            <xs:element ref="hd:toggle" />

                        <xs:element ref="m2m:subscription"  />
                    </xs:choice>
                </xs:choice>

            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

```
1  package ule.hanfun.interfaces;
2
3  public interface AlertServerInterface {
4
5      // Actions
6
7      void status(Integer unitType, Integer stateAttribute);
8
9      // DataPoint getters/setters
10
11     Integer _getState();
12     void _setEnable(Integer enable);
13     Integer _getEnable();
14 }
```

Work on Information Models

- Soon after the initial launch of release 1.0 of oneM2M, member companies requested to develop information models that can be interworked from various IoT technologies and represented to applications with a consistent fashion.
  - The work of the Proximal IoT Interworking specification (TS-0033) that incorporates various interworking technologies (e.g., LwM2M, OCF, AllJoyn, DDS, OPC-UA, WoT, Modbus)
  - The work of the generic interworking mechanism (TS-0030) is to specify a procedure where devices do not necessarily require a specific representation (e.g., KNX).
  - The first domain requesting development of information models was the home domain – this led to a Study of the Home Domain (TR-0013) and follow-on device model specifications (TS-0023).
- Based on the successful work with the Home Appliance models, member companies are looking toward developing information models in the Industrial (TR-0018), Vehicular (TR-0026) and Smart City (TR-0046) Domains. But we will only do these in collaboration with experts in these domains.

Home Appliances Models

- Based on SDT, oneM2M defines **Information Models for Home Appliances** in TS-0023 and its latest version is v.3.4.0.
- These appliance/device models were contributed by device manufacturers, service providers as well as adaptations from other standards (e.g., ECHONET, OMA Device WebAPIs)
- This version currently includes 42 device models and 70 module classes
- TS-0023 is still evolving and has more device models and functionality for Rel-3
- Television device model example

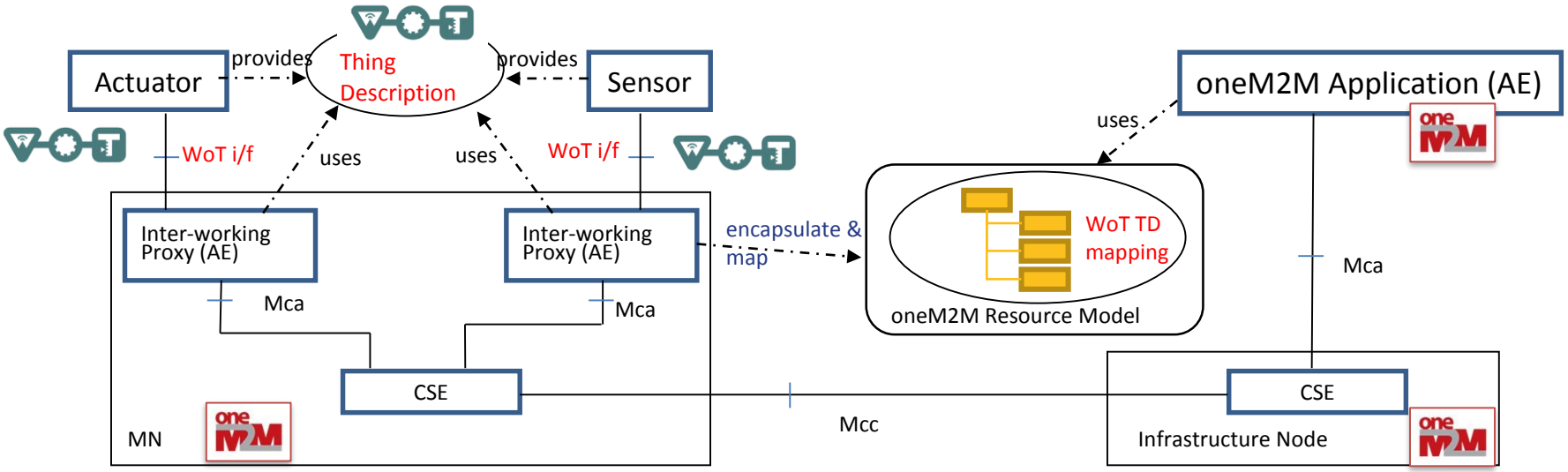
	Properties (Optionality)	**Module Class** (Optionality)	**Actions** (All optional) **& Data Points** (Optionality)
TV	manufacturer(M)	binarySwitch(M)	toggle(), powerState(O)
	modelID(M)	audioVolume(O)	upVolume(), downVolume(), volumePercentage(M), stepValue(O), muteEnbled(M),…
	deviceType(M)	televisionChannel(O)	upChannel(), downChannel(), chNumber(M), availableCh(O), perviousCh(O), chName(O)
	deviceName(O)	playerControl(O)	nextTrack(), previousTrack(), currentPlyerMode(M), modeName(O), modes(M), speed(O)
	country(O)	mediaInput(O)	mediaID, supportedMediaSources, mediaName, status, mediaType
	swVersion(O)	mediaOutput(O)	mediaID, supportedMediaSources, mediaName, status, mediaType

Home Appliances Models: Semantic Interoperability

- Realizing the need for alignment of information/data models as a first step in interoperability, oneM2M is starting a process of aligning its models with other groups (e.g., OCF). Remember that the original models were, in part, based on work of other standards organizations.
- As the first target, oneM2M recently has started the process to align Home Appliance information models with OCF's version 1.0 data models.
  - 25 of 35 devices were only defined in OCF 1.0 while 10 were defined in both
  - 25 new devices are added to oneM2M, and 10 existing oneM2M models are modified to be aligned with OCF 1.0 data models
  - So now 35 devices in OCF 1.0 is all mappable with oneM2M devices
- Next step for OCF data model mapping
  - OCF may need to define new device models that are existing in only oneM2M to support full interoperability in the future

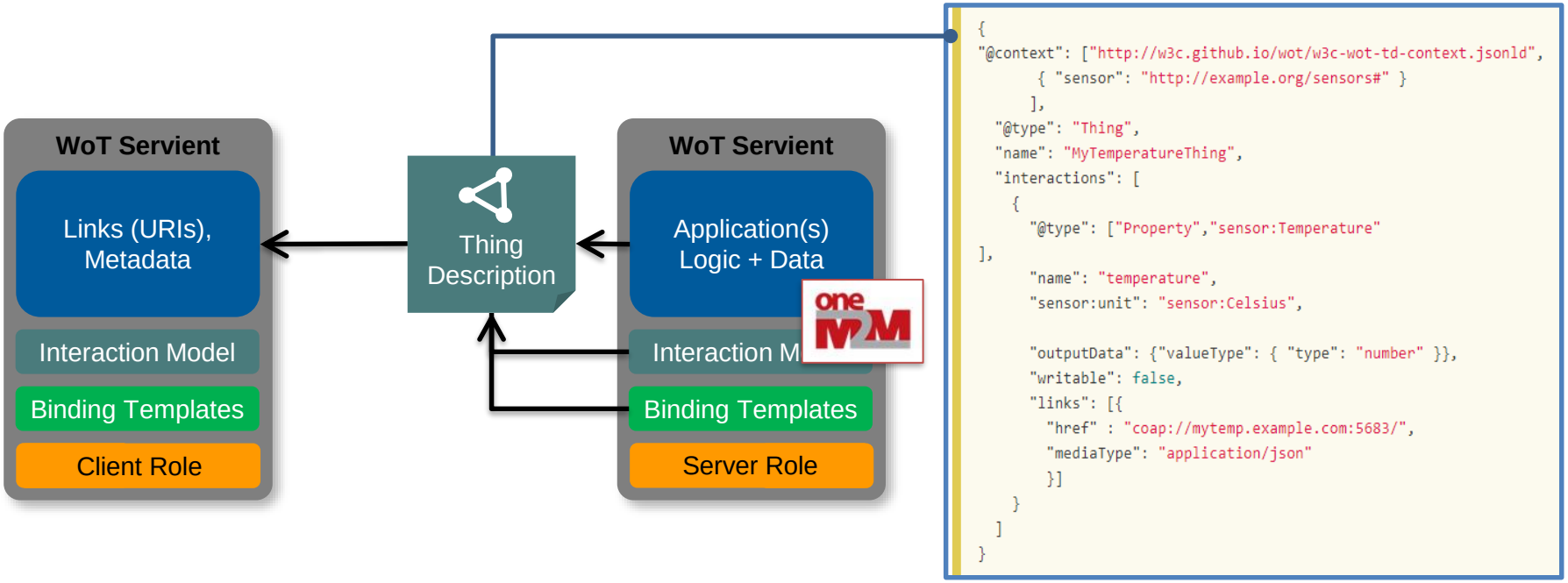
Example: WoT Interworking – WoT Interface

- Exposing the WoT interface (described in TD) to oneM2M systems
  - Benefit: WoT services/data can be consumed by oneM2M applications

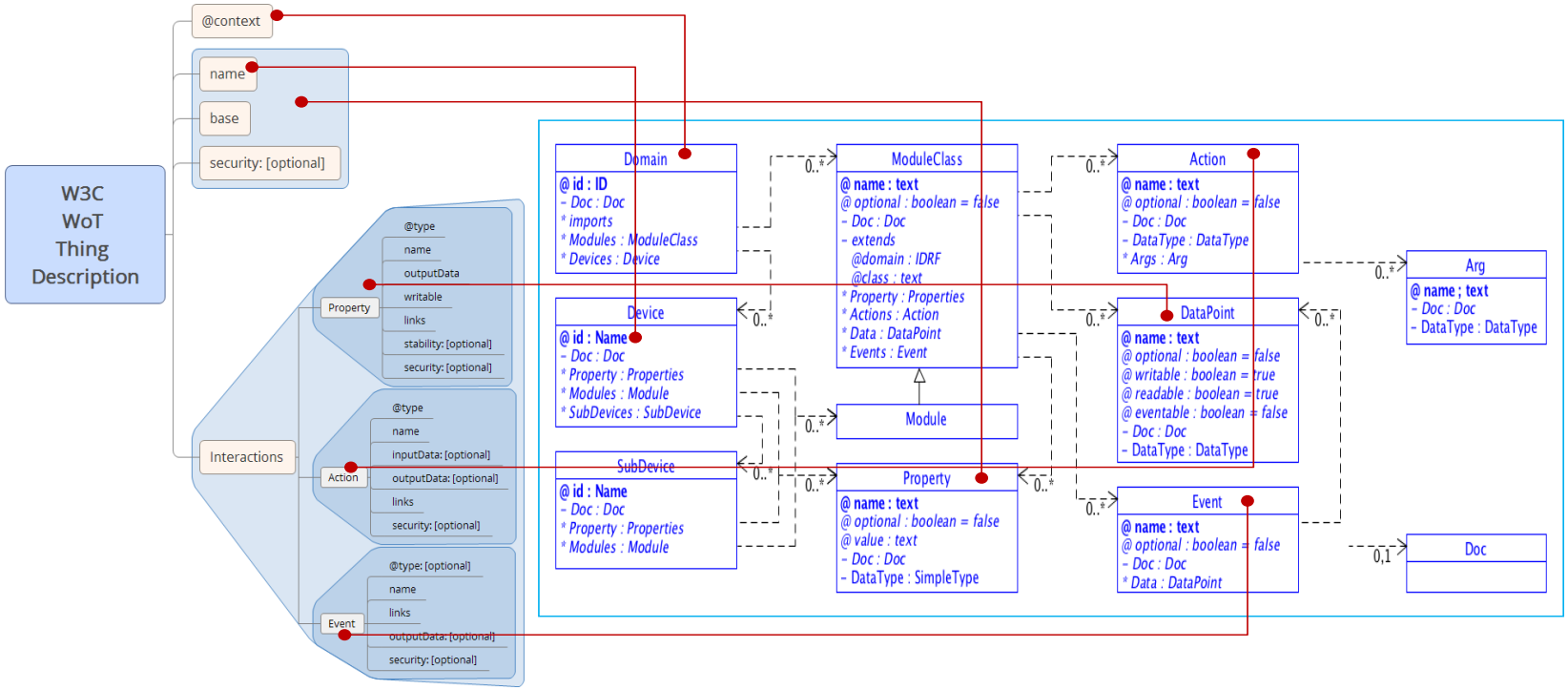


Example: WoT Interworking – oneM2M Interface

- Exposing oneM2M interfaces to WoT systems
  - Benefit: oneM2M services/data can be consumed by WoT Servients

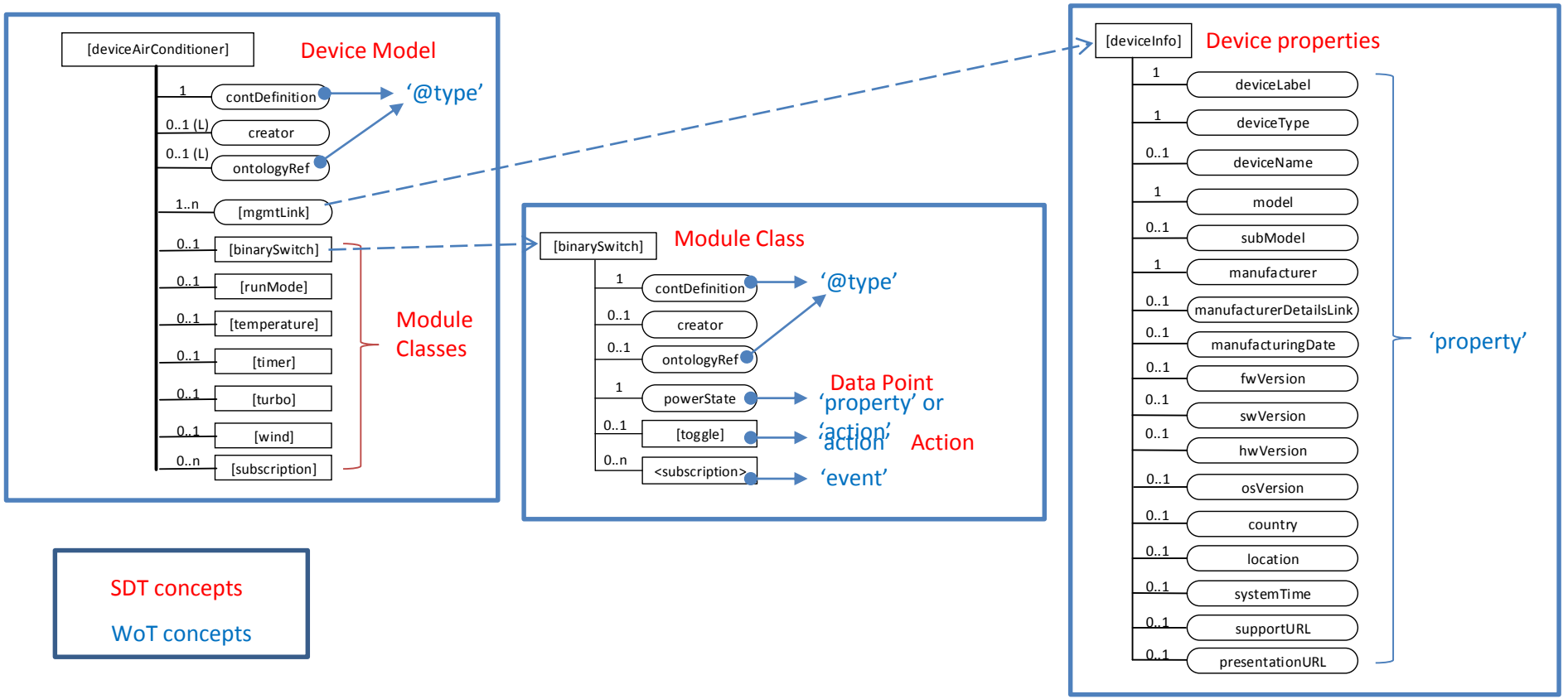


Example: WoT Interworking - SDT Mapping



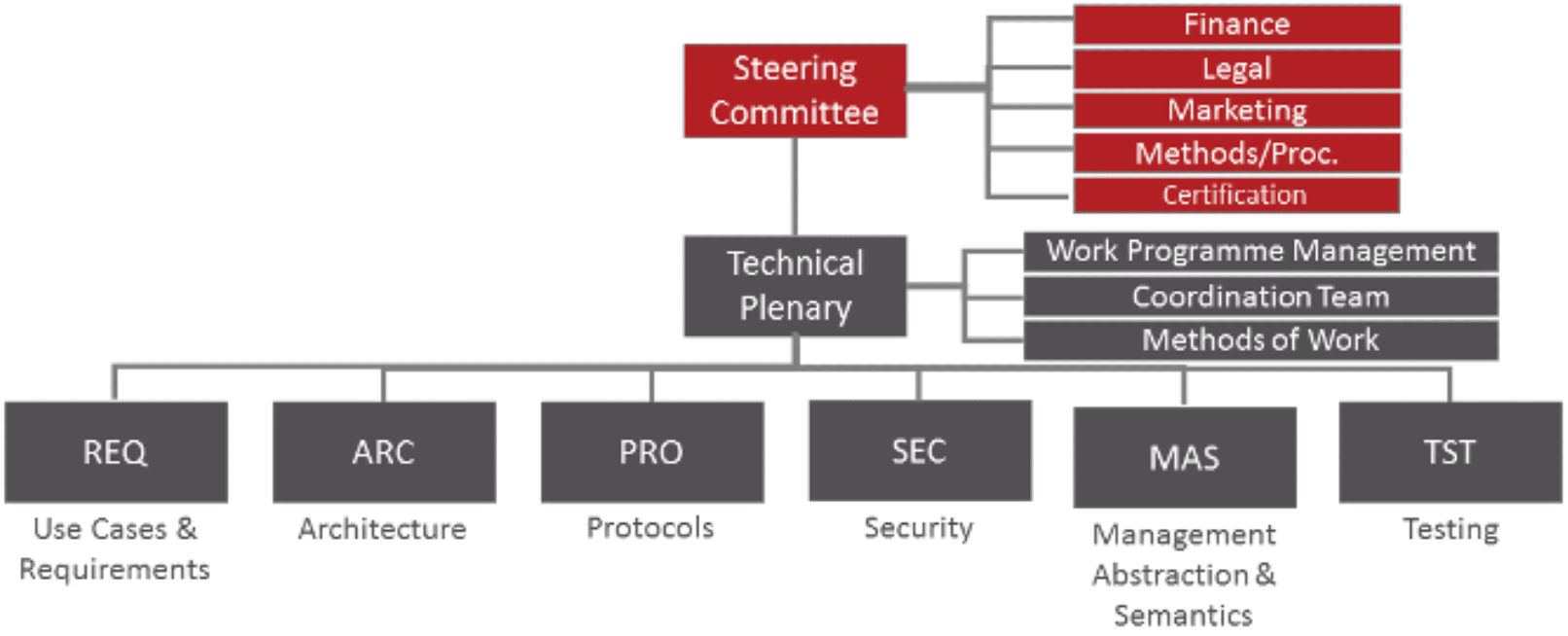
- Conceptual alignment observed between oneM2M/HGI SDT and W3C WoT information models
- It implies promising semantic interoperability.
- Technical study is progressing on both sides

Example: WoT Interworking – Home Automation with SDT



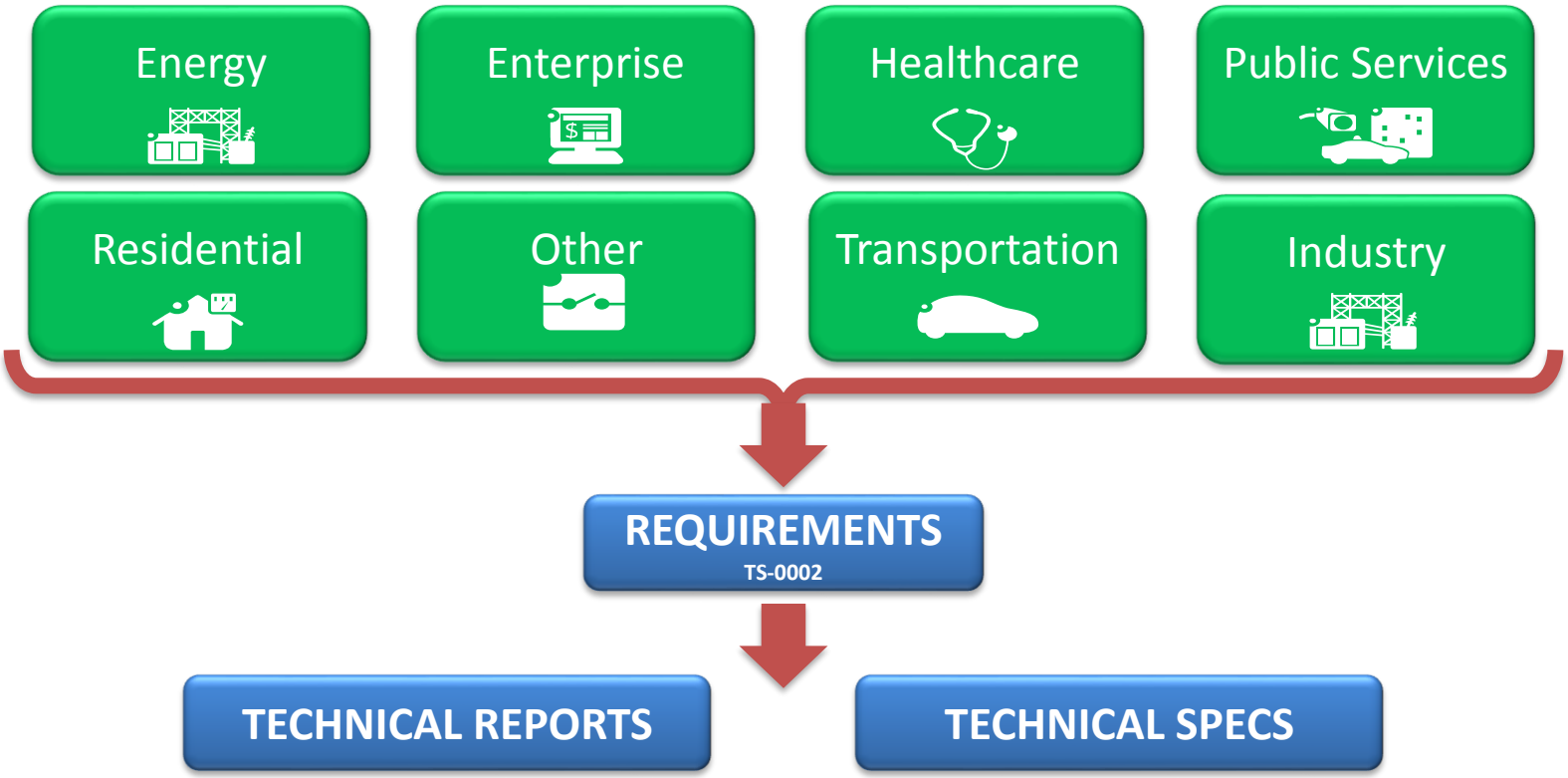
How do you work?

- oneM2M is a partner project standards organization so transparency and consensus is at the core of the processes.



How do you work?

- Typically features are developed through a study of the feature which leads to requirements that are applied to the architecture which is then implemented in a set of specifications.



How do you work?

- Once the specifications are approved, features are then available for commercial and open source implementations to come together in order verify the specification set. The interop events, with formal test specifications, typically occur 2-3 times a year around the world.
- The results from these interop events are then feed back into oneM2M as a series of change requests.

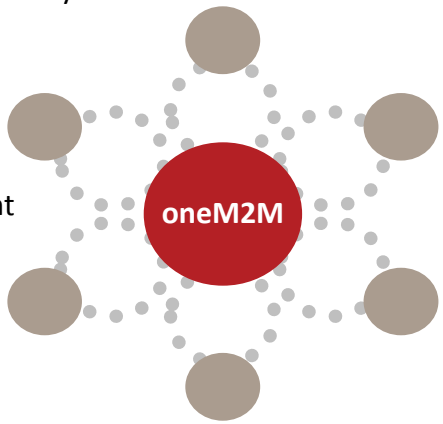
- In addition to the feedback from interop events, oneM2M also has a feedback mechanism from any interested party to submit a problem via the oneM2M developer e-mail list: TechQuestions@list.onem2m.org

How far did you get?

- oneM2M is currently is working on its 3rd formal release. Release 1.0 was published in 2015; release 2.0 in 2016; release 3.0 is expected at the end of 2017. An example of the release 3.0 functionality show the focus on Interoperability (interworking and semantics), Market Adoption and Additional security and privacy features

Configuration and Enrollment Management
- M2M Application & Field Domain Component Configuration
- Enrollment & Authentication APIs
- Interoperable Privacy Profiles

Security
- Distributed Authorization
- Decentralized Authentication
- Secure Environment Abstraction and public key framework
- automated certificate enrollment



Market Adoption
- additional Developer Guides
- oneM2M Conformance Test
- Feature Catalogues
- Product Profiles
- Smart Cities

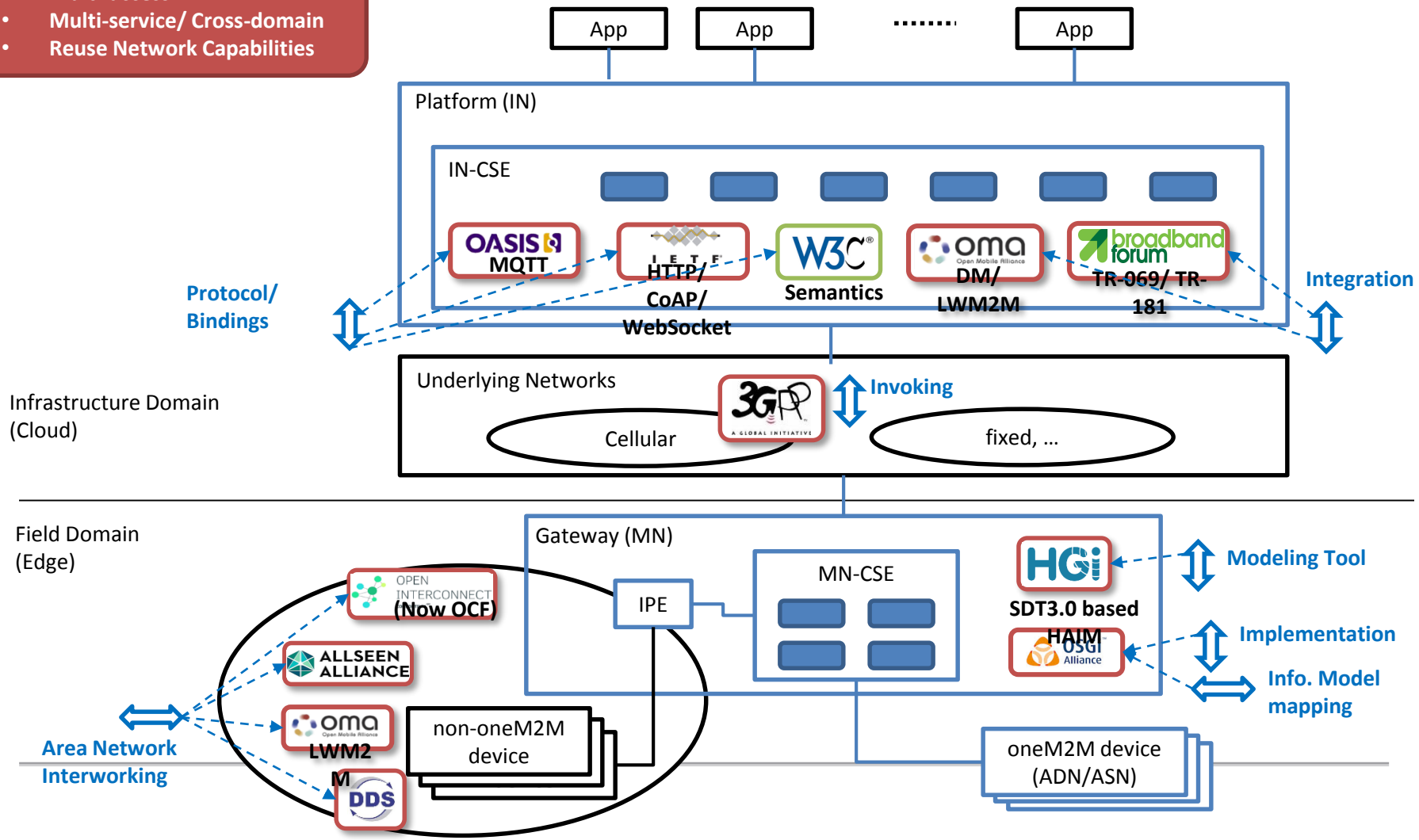
Interoperable with
- 3GPP SCEF
- OMA LWM2M
- DDS
- OPC-UA
- Modbus
- Proximal IoT
- OSGi
- W3C WoT

Features
- Atomic Transactions
- Action Triggering
- Cross resource subscriptions
- Streaming support
- Heterogeneous Identification

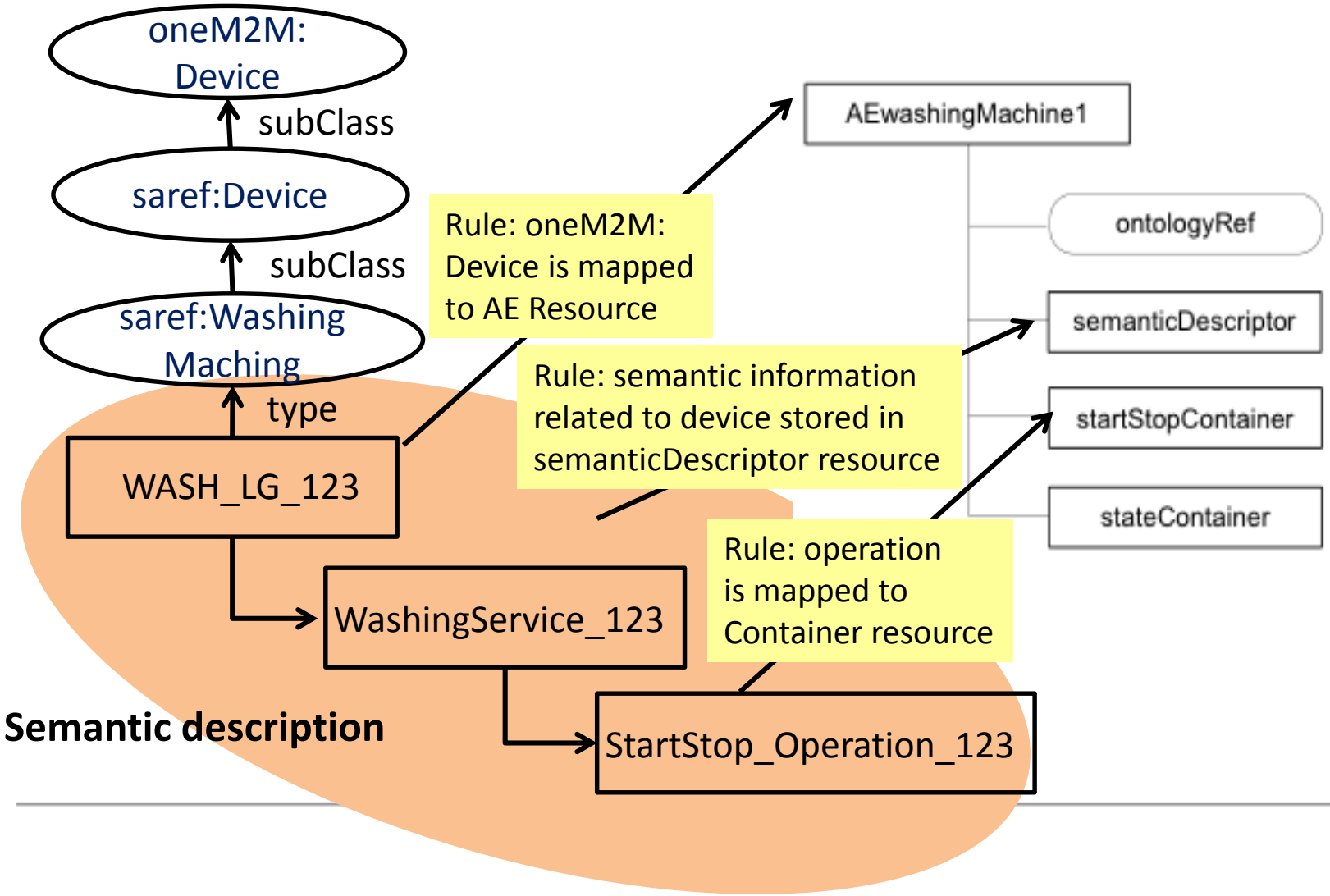
Semantics and Abstraction
- Semantic Querying
- Semantic Mashups
- Home Domain Information Model enhancements

How can your work be integrated into other work: Through Interworking

- Multi-access
- Multi-service/ Cross-domain
- Reuse Network Capabilities



How can your work be integrated into other work: Through Ontology subclassing



How can other organization work together with you? – Current Collaborations

- ✓ Information Sharing (Liaison, workshop, …)
  - SG20
  - JTC1 WG10
  - P2413
  - Semantics, WoT
  - Connected Living
  - Certification
  - Interworking
  - OIC Interworking (Now OCF)
  - AllJoyn Interworking
  - Network Service Exposure
  - SCP, SmartM2M
  - ref. arch.
- ✓ Endorsement (adoption)
  - MQTT
  - HTTP/ CoAP/ (D)TLS/ WebSocket
  - DDS
  - TR-069/ TR-181
  - OMA DM/ LWM2M
- ✓ Partnership
- ✓ Merge/ Integration

How can other organization work together with you? – Process and Legal Issues

- oneM2M specifications are open for review by any organization.
- Typically communication is done through a set of liaisons, typical for standards organizations.
- In some cases oneM2M specifications are republished by other organizations. In this case, the oneM2M legal and steering committee draw up an agreement that is amenable to both parties.

How can research contribute to the success of your work?

- With over 200 members, oneM2M has research organizations that are member partners of oneM2M. These research organizations have in the past brought features into the specification (e.g., Semantics, IoT Security).

- Research organizations can also choose to be an Associate Member that has the right to attend and provide input to meetings of the Technical Plenary and its subgroups. However, this is limited to clarifications regarding regulatory matters and informational contributions.

- For non-members, research organizations have the same access as other developers to comment and ask questions but only oneM2M partners are able to contribute to specifications.

- Membership is easy! http://www.onem2m.org/membership/join-onem2m