# oneM2M Semantic enablement and ASD (Advanced Semantic Discovery) for (AE) "Resources"

## oneM2M Semantic enablement and ASD (Advanced Semantic Discovery) for (AE) "Resources"

The oneM2M Architectural Model in oneM2M  Semantic enablement specification is based on the generic oneM2M Architecture for the Common Service Layer specified in oneM2M TS Functional Architecture.

The Core Functionality supporting Semantics resides at various CSEs, providing Services to the AEs via the Mca Reference Point and interacting with other CSEs via the Mcc Reference Point.

The Semantics (SEM) CSF (Common Service Function) is an oneM2M Common Service Function (CSF) which enables Semantic Information Management (SIM) and provides the related functionality based on this Semantic Information. The Functionality of this CSF is based on Semantic descriptions and implemented through the specialized resources and procedures described in oneM2M Semantic Enablement specification

The SEM CSF includes  Specialized Functional Blocks such as: SPARQL Engine, Repositories for Ontologies and Semantic Descriptions, which may be implemented via Permanent or Temporary Semantic Graph Stores, etc.
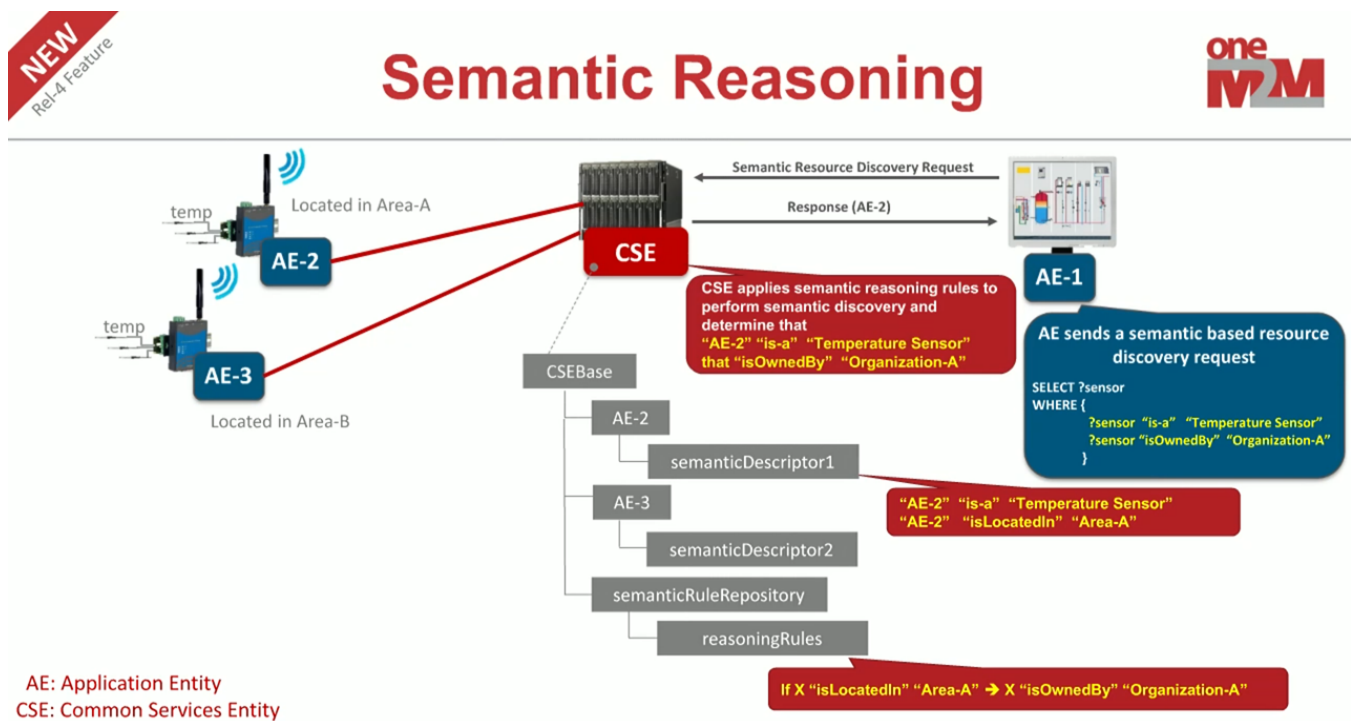
# 6 Description of Classes and Properties

## 6.1 Classes

### 6.1.1 Class: Thing

## Class: Thing



**Figure 2: Thing**

**Description**

- A **Thing** in oneM2M (Class: Thing) is an entity that can be identified in the oneM2M System.
  A Thing that is not a Device is not able to communicate electronically with its environment. However, the sub-class of Thing that is able to interact electronically is called a "Device".
  A Thing may have ThingProperties (Object Property: hasThingProperty). A Thing can have relations to other things (Object Property: hasThingRelation).
  Since a Thing that is not a Device is not able to communicate electronically it cannot influence the value of its ThingProperties or being influenced by it. Similarly a Thing cannot document its - real-world - relationships (via hasThingRelation) to other Things.

### 6.1.2 Class: ThingProperty

## Class: ThingProperty



**Figure 3: ThingProperty**

**Description**

- A **ThingProperty** (Class: ThingProperty) denotes a property of a Thing. A ThingProperty can e.g. be observed or influenced by devices, or it constitutes static data about a Thing.
  E.g. the indoor temperature of the room could be a ThingProperty of a Thing "room".
  A ThingProperty of a thing can describe a certain Aspect, e.g. the indoor temperature describes the Aspect "Temperature" that could be measured by a temperature sensor.
  A ThingProperty of a Thing can have meta data.

- The class ThingProperty is a sub-class of the Variable class.

**Object Properties**

This Class is the domain Class of Object Property:

- describes (range Class: Aspect)
  (inherited from class: Variable)

### 6.1.3 Class: Aspect

## Class: Aspect



**Figure 4: Aspect**

**Description**

- An **Aspect** (Class: Aspect) describes the real-world aspect that a Function relates to. Aspect is also used to describe the quality or kind of a Variable.
  The Aspect could be a (physical or non-physical) entity or it could be a quality.
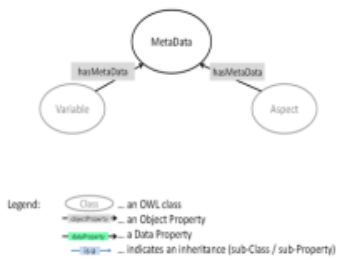
## 6.1.4 Class: MetaData

## Class: MetaData



**Figure 5: MetaData**

**Description**

- **MetaData** (Class: MetaData) contain data (like units, precision-ranges ...) about a Variable or about an Aspect.

  E.g. the indoor temperature could have as meta data an individual "Celsius_Scale" that specifies that the temperature needs to be understood as degrees Celsius.

**Object Properties**

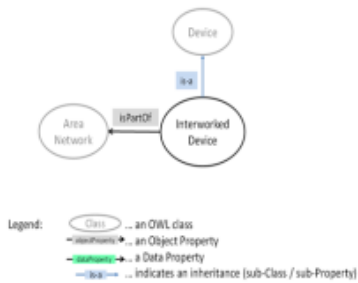## 6.1.6 Class: InterworkedDevice

## Class: InterworkedDevice



**Figure 7: InterworkedDevice**

**Description**

- An **InterworkedDevice** (Class: InterworkedDevice) is a Device - e.g. in an Area Network - that does not support oneM2M interfaces and can only be accessed from the oneM2M System by communicating with a "proxied" (virtual) device that has been created by an Interworking Proxy Entity.

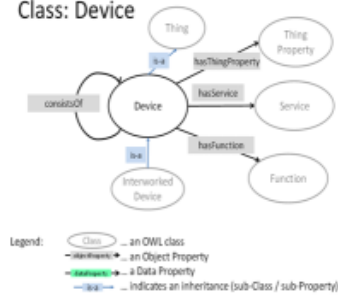## 6.1.5 Class: Device

## Class: Device



**Figure 6: Device**

**Description**

- A **Device** (Class: Device) is a Thing (a sub-class of class:Thing) that is designed to accomplish a particular task via the Functions the Device performs.

  A Device can be able to interact electronically with its environment via a network. A Device contains some logic and is producer and/or consumer of data that are exchanged via its Services with other oneM2M entities (Devices, Things) in the network. A Device may be a physical or non-physical entity.

  A Device interacts through the DataPoints and/or Operations of its Services:

### 6.1.7    Class: AreaNetwork

## Class: AreaNetwork



**Figure 8: AreaNetwork**

**Description**

- An **AreaNetwork** (Class: AreaNetwork) is a Network that provides data transport services between an Interworked Device and the oneM2M System. Different area Networks can use heterogeneous network technologies that may or may not support IP access.

### 6.1.9    Class: Function

### 6.1.9.0    General description

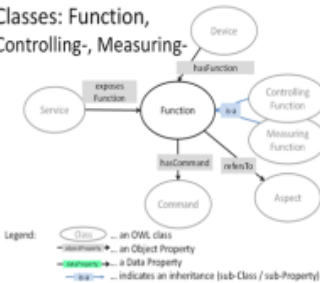## Classes: Function, Controlling-, Measuring-



**Figure 10: Function**

**Description**

- A **Function** (Class: Function) represents a particular function necessary to accomplish the task for which a Device is designed. A device can be designed to perform more than one Function.
  The Function exhibits the - human understandable - meaning what the device "*does*".

- A Function refers to (e.g. observes or influences) some real-world aspect(s), that can be modelled as a Class: Aspect.

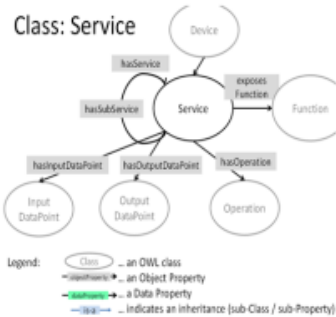### 6.1.8    Class: Service

## Class: Service



**Figure 9: Service**

**Description**

- A **Service** (Class: Service) is an electronic representation of a Function in a network. The Service exposes the Function to the network and makes it discoverable, registerable and remotely controllable in the network.
  A Service is offered by a device that wants (a certain set of) its Functions to be discoverable, registerable, remotely controllable by other devices in the network.
  A Service can expose one or more Functions and a Function can be exposed by one or more Services.

6.1.10.0    General description



Figure 11: Operation
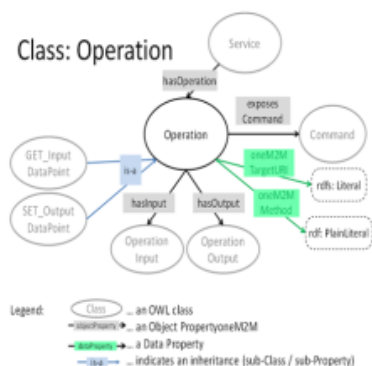
6.1.16    Class: Variable

6.1.16.0    General description



Figure 17: Variable

**Description**

- A **Variable** (Class: Variable) constitutes a super class to the following classes: ThingProperty, OperationInput, OperationOutput, InputDataPoint, OutputDataPoint. Additionally, class:Variable is the disjoint union of classes: SimpleTypeVariable and StructuredTypeVariable, i.e. any member of class:Variable is also member of either SimpleTypeVariable or StructuredTypeVariable.
The members of class:Variable are entities that store some data (e.g. integers, text, etc., or structured data) that can change over time. These data of the Variable usually describe some real-world Aspects (e.g. a temperature) and can have MetaData (e.g. units, precision, etc.).

6.1.16.2    Class: StructuredTypeVariable



Figure 19: Variable
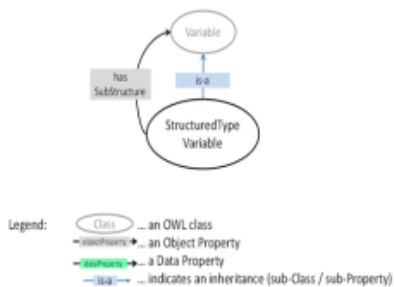
## oneM2M Semantic enablement  for (AEs) and ETSI Smart M2M Resource ASD (Advanced Semantic Discovery)

## Semantic Discovery in presence of a "Network" of M2M Service Providers (M2MSPs)

The Advanced Semantic Discovery (ASD) aims to discover AEs (also called Resources) that are registered/announced to some CSEs.

The ASD could start from any AE, even these ones not belonging to the same Trusted Domain.

The ASD differs from the usual one present in oneM2M in the sense that one (or many) AE could be searched for even without knowing its identifier, but just knowing its TYPE or ONTOLOGY membership, as shown in Figure 6.3.1-1.
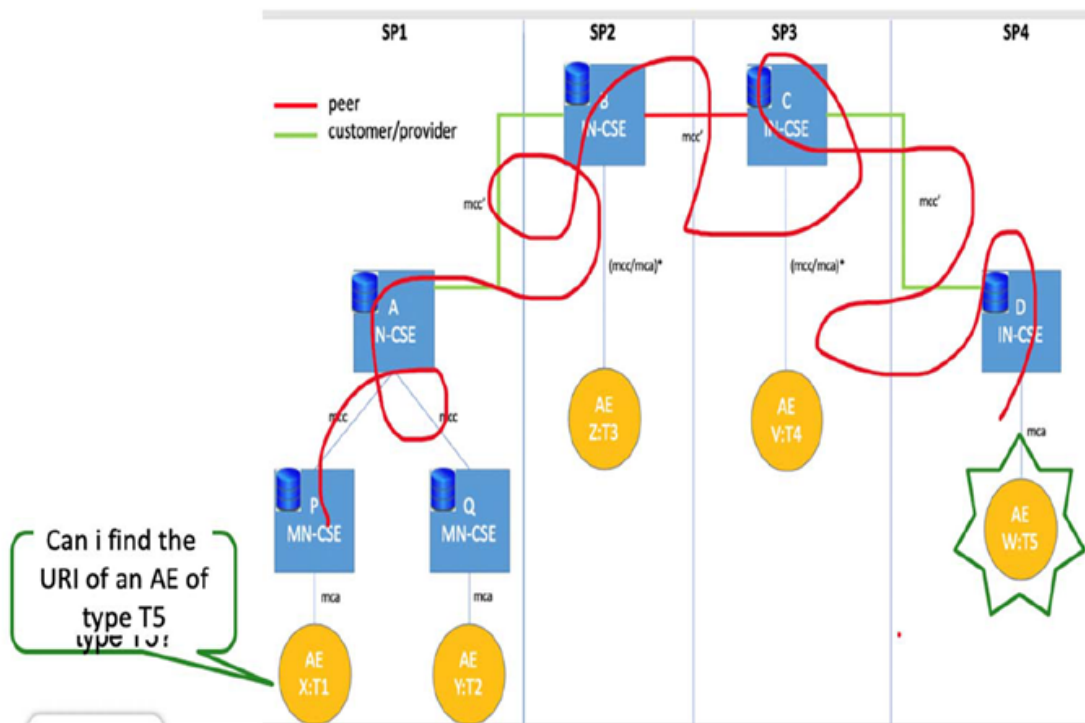
Fig. 6.3.1-1 Advanced Semantic Discovery (ASD) in one image

Advanced Semantic Discovery (ASD) in Figure 6.3.2-1 below describes oneM2M as Semantic Discovery involving multiple CSEs.
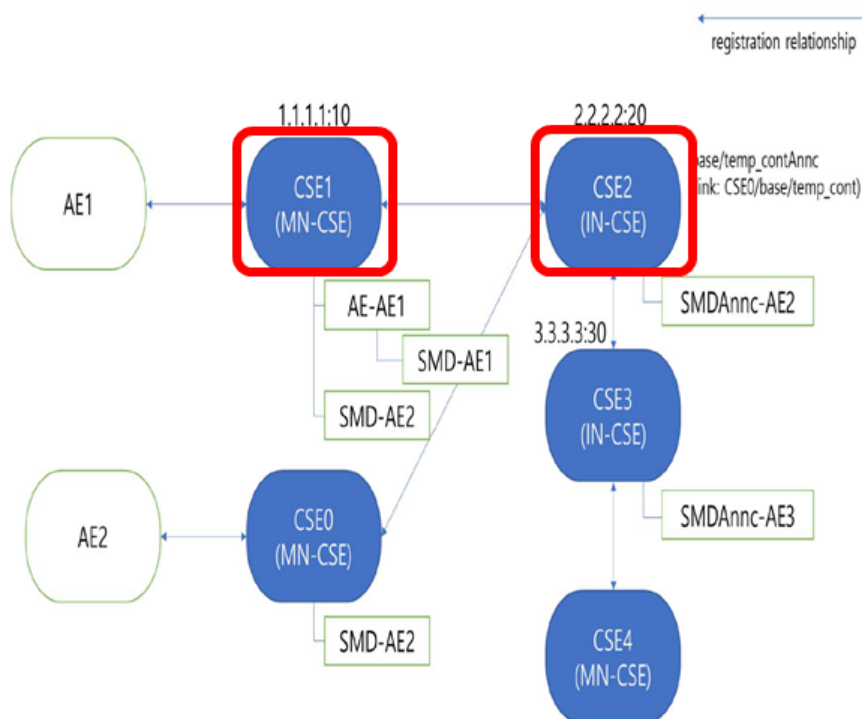


Figure 6.3.2-1: Actual oneM2M limited discovery routing in one slide

**ASD within Distributed Network of CSEs belonging a single Service Provider & across different IoT Service Providers.**
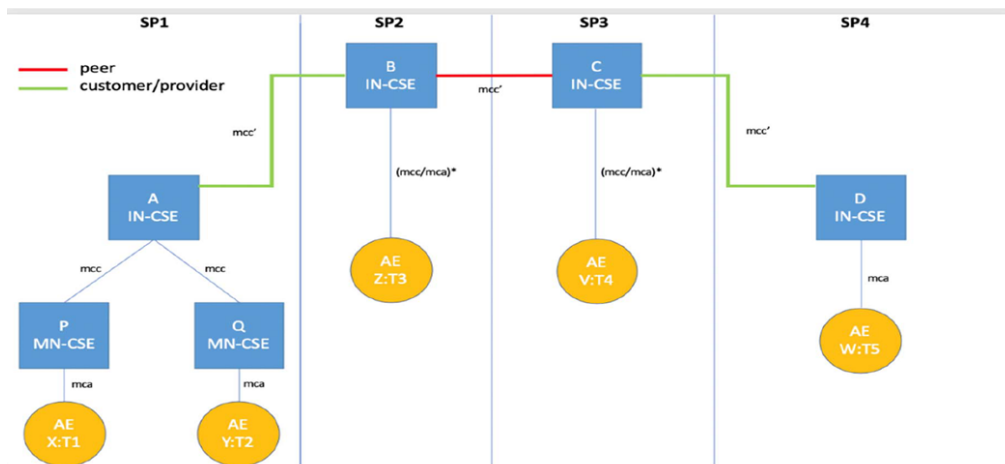
Figure 5.4-1: Pre-condition topology

**Example of SRT (Semantic Routing Table)**



Table 7.2.6-2: Upgrading the adjacent SRT CSEs with the new y AE-THERMOMETERS

| CSEcust1 | | | | |
|---|---|---|---|---|
| TYPE | URI | CSE CUSTOMERS | CSE PEERS | CSE PROVIDERS |
| THERMOMETER | URI_v, URI_w | .. | .. | (CSE, #_+y) |

| CSEcust2 | | | | |
|---|---|---|---|---|
| TYPE | URI | CSE CUSTOMERS | CSE PEERS | CSE PROVIDERS |
| THERMOMETER | URI_z | .. | .. | (CSE, #_+y) |

| CSEpeer | | | | |
|---|---|---|---|---|
| TYPE | URI | CSE CUSTOMERS | CSE PEERS | CSE PROVIDERS |
| THERMOMETER | URI_a | .. | (CSE, #_+y) | .. |

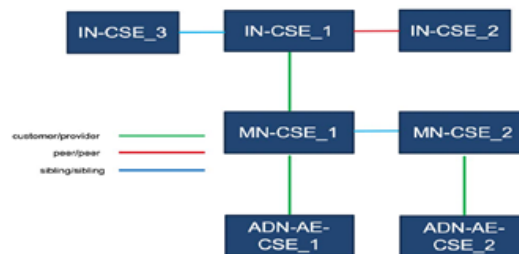| CSEprov | | | | |
|---|---|---|---|---|
| TYPE | URI | CSE CUSTOMERS | CSE PEERS | CSE PROVIDERS |
| THERMOMETER | URI_b, URI_c, URI_d | (CSE, #_+y) | .. | .. |

Figure 7.2.1-1: C2P and P2P and S2S CSE relationships

The Ontology Mapping Task performed by

=> Create Operation or

=> Update Operation against an *<OntologyMapping> R*esource on a Hosting CSE.

A Retrieve operation against the same *<OntologyMapping>* Resource shall be used to get the result of Ontology mapping. A Delete operation against a *<OntologyMapping> R*esource shall follow the basic procedure as specified.
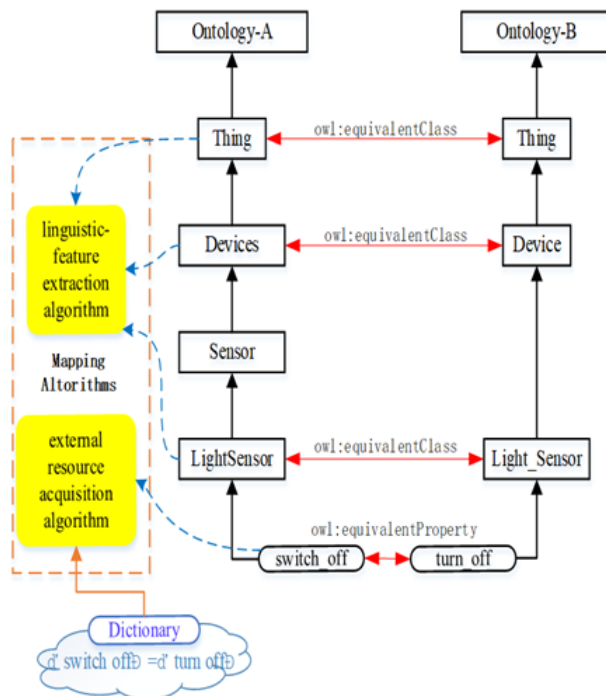
Figure 6.10.2-2: Example of the mapping result between ontology A and ontology B

## 3GPP 5G NDL (Network Data Layer) and oneM2M Semantic enablement and ETSI SmartM2M ASD integration

The information related to oneM2M Semantic enablement and ETSI SmartM2M ASD support (integration) with 3GPP specified 5G NDL (Network Data Layer in which Data "Compute" is separated from "Storage" in the process of virtualization of 5G NFs into VNFs/PNFs by separating the context in the NF's Application related Data from the Business Logic in the NF's Application related Data and stored separately in Nodes specified by 3GPP for 5G and denoted as "Structured" and "Unstructured" Data and supported in 5G 3GPP Rel 16 ATSSS (Access Traffic Steering, Switching and Splitting) is deliberately not included and part of the presentation on the oneM2M and 5G New Services.
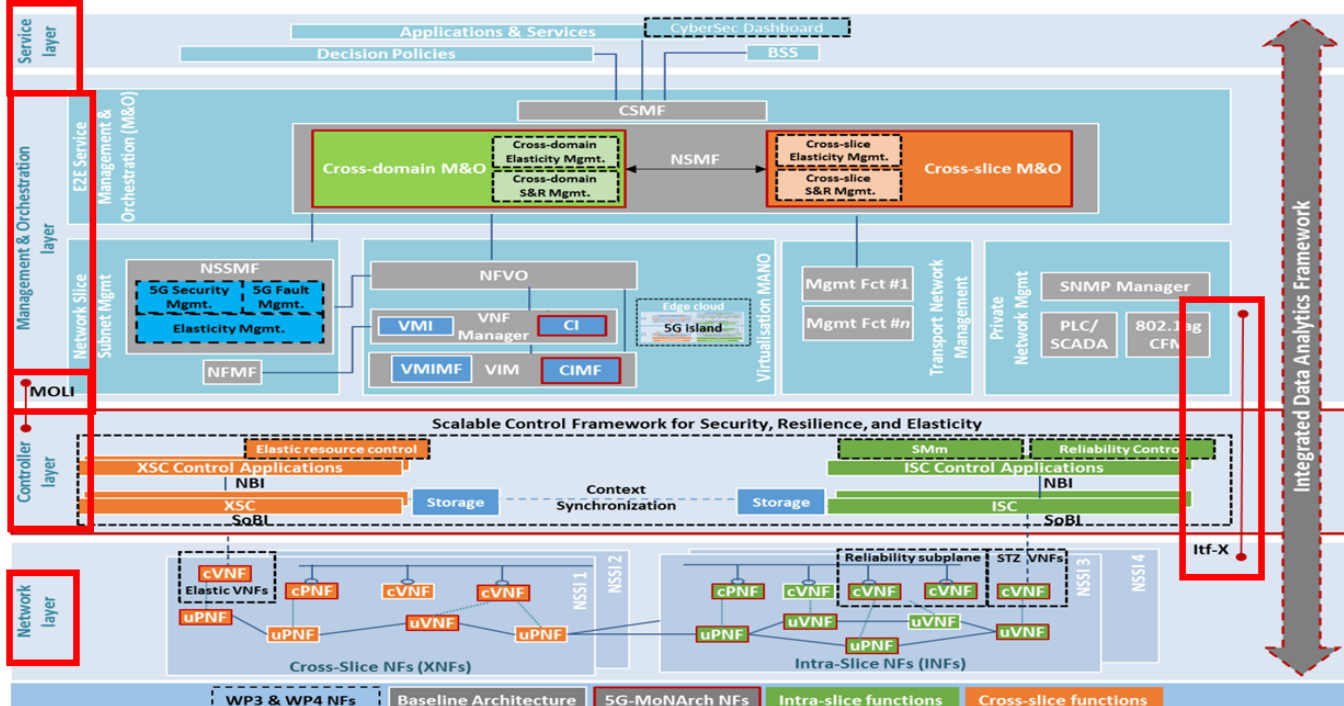


Figure 2-2: 5G Mobile Network overall Architecture