Service Layer API for oneM2M

Draft

93 Pages

Text in Red is here to help you. Delete it when you have followed the instructions.

The <RFC Title> can be set from the File>Properties:User Defined menu. To update it onscreen, press F9. To update all of the fields in the document Select All (CTRL-A), then hit F9. Set the release level by selecting one from: Draft, Final Draft, Release. The date is set automatically when the document is saved.

Abstract

10 point Arial Centered.

oneM2M is standard organization and specifies middleware for IoT, called Common Services Entities (CSE). Application can access functionality in CSE with RESTful operations, which are Create, Retrieve, Update, Delete and Notify. oneM2M allows variety of communication methods, 4 protocol bindings (HTTP, MQTT, CoAP, Websocket) and 3 serializations (XML, JSON, CBOR). This RFC describes the way to provide high level API for oneM2M RESTful operations hiding the difference of variety of communication methods.

0 Document Information

0.1 License

DISTRIBUTION AND FEEDBACK LICENSE, Version 2.0

0.2 Trademarks

0.3 Feedback

This document can be downloaded from the OSGi Alliance design repository at https://github.com/osgi/design The public can provide feedback about this document by opening a bug at https://www.osgi.org/bugzilla/.

0.4 Table of Contents

0.5 Terminology and Document Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY" and "OPTIONAL" in this document are to be interpreted as described in 10.1.

```
Source code is shown in this typeface.
```

0.6 Revision History

The last named individual in this history is currently responsible for this document.

Revision	Date	Comments
Initial	*SEP 15 2017*	Initial Contribution. Hiroyuki Maeomichi, NTT, maeomichi.hiroyuki@lab.ntt.co.jp
0.0.1	*SEP 21 2017*	Updated alternatives, some figures, added description on validator. Hiroyuki Maeomichi, NTT, maeomichi.hiroyuki@lab.ntt.co.jp

Revision	Date	Comments
0.0.2	*April 17 2018*	Update based on discussion in Washington meeting. Hiroyuki Maeomichi, NTT, maeomichi.hiroyuki@lab.ntt.co.jp
0.0.3	*June 22 2018*	Add new fields and class reflecting R3 draft of oneM2M: Added fields in RequestPrimitiveDTO, ResponsePrimitiveDTO, and FilterCriteriaDTO, and ReleaseVersion enum. Organize DTOs: Added AttributeDTO, LocalIdTokenIdAssigmentDTO, and DasInfoDTO and remove DynAuthLocalIdAssignmentsDTO and DynAuthReqInfoDTO Introduce OperationIF interface as a super interface of ProtocolBinding interface and CSE interface for enabling concise application code. This replaces former simple.Client. Organize Introspection interfaces with less methods. They are moved to dedicated package.
0.0.4	*June 25 2018*	Add section ‘Mapping Rules for Generic DTO’ Update Javadoc with more explanations. (moved old classes to org.osgi.service.onem2m.old package for preparing deletion.) Add description to Security Consideration section. Add oneM2M R3 specs and XSD to references.
0.0.5	*Jun 27 2018*	Modified after discussion in Washington DC F2F. Restructure service interfaces; now 2 interface is remaining. For receiving notification, dedicated interface is prepared. Remove Introspection interfaces. Reduce service properties by removing ones for informative purpose. Add example flow to control devices. Modify security consideration in section 9.2.
0.06	*Jun 29 2018*	Add examples, with code snipets. Add ‘Data Modification in Protocol Binding’ section. Add discovery() method with additional parameter.
0.07	*Sep 17 2018*	Remove all service properties. Add some reasons to considered alternative section.

1 Introduction

Introduce the RFC. Discuss the origins and status of the RFC and list any open items to do.

oneM2M is standard organization and specifies middleware for Internet of Things (IoT), called Common Services Entities (CSE). Applications can access CSE's functionality with RESTful operations, which are Create, Retrieve, Update, Delete and Notify. TS-0001 [2] defines more than 40 resource types to expose CSE's functionalities. oneM2M allows variety of communication methods, combination of 4 protocol bindings (HTTP, MQTT, CoAP, Websocket) and 3 serializations (XML, JSON, CBOR).

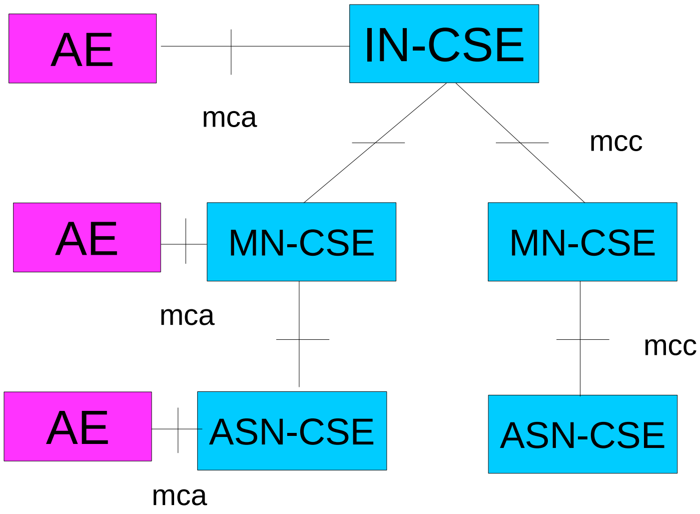
This RFP discuss the way to provide high level API (namely service layer API) for oneM2M RESTful operations hiding the difference of variety of communication methods.

2 Application Domain

This section should be copied from the appropriate RFP(s). It is repeated here so it can be extended while the RFC authors learn more subtle details.

2.1 IoT Application configuration using oneM2M

oneM2M's middleware, called CSE can be deployed in different locations and they are connected each other forming tree topology. Depending on deployed location, CSEs are categorized to 3 types, IN-CSE, MN-CSE and ASN-CSE. IN-CSE is located top of tree, ASN-CSE is located at leaf and MN-CSE is located and MN-CSE is located on middle.

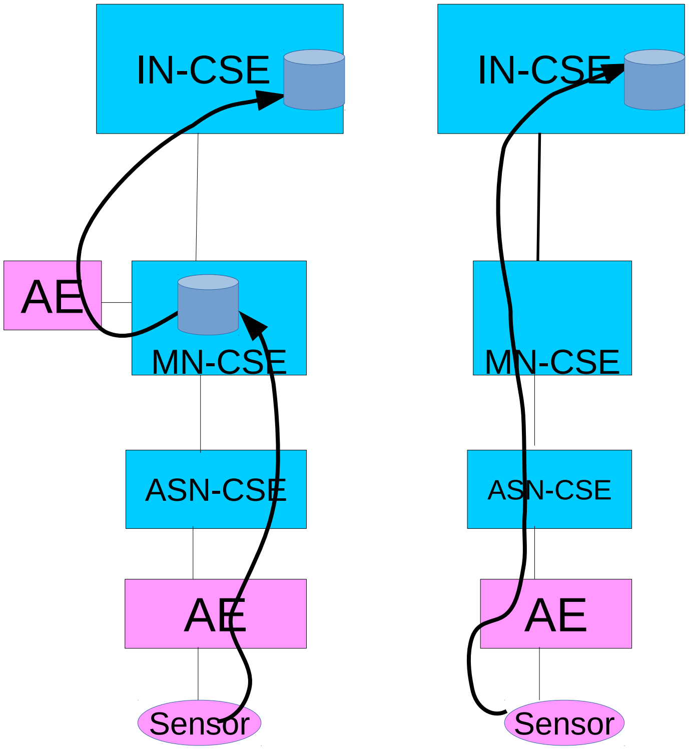


oneM2M’s application, called Application Entity (AE) connects to one of CSEs. After AE connecting to the CSE, AE can access to all of CSEs, by retargeting function (similar to routing) of CSEs.

AE accesses to CSE’s functionality through RESTful API, which consists of Create, Retrieve, Update, Delete and Notify in targeting more than 40 types of resources. For examples, typical resources are <*contentInstance*> that expresses IoT data and <c*ontainer*> that holds set of <*contentInstance*>s. AE can create or retrieve the <contentInstance> on any CSE by the retargeting functionality, as far as permission is allowed. Interface between CSEs is called *mcc* and interface between CSE and AE is called *mca,* both interfaces have almost same interface.

It is possible to develop variety types of distributed applications using the architecture. For example for IoT data aggregation applications, it is possible to develop gradual aggregation type or direct aggregation type. In gradual aggregation type, AE connected to ASN-CSE creates <*conentInstance*>s in ASN-CSE, and intermediate applications calculate statistics and put the result on IN-CSE as a <*contentInstance*>, while, in direct aggregation type, AE connected to ASN-CSE creates <*contentInstance*>s in IN-CSE directly.

Under CSE layer, oneM2M specifies NSE(Network Services Entity), but this RFC doesn’t cover the NSE layer.



2.2 Communication methods used in oneM2M

oneM2M allows variety of communication methods, combination of 4 protocol bindings (HTTP, MQTT, CoAP, Websocket) and 3 serializations (XML, JSON, CBOR). It might be added in future. oneM2M specifies specification in different level.

Firstly TS-0001[2] specifies high level resource definitions, it defines more than 40 resource types, such as <contentInstance> for storing IoT data, <timeSeriesInstance> for periodic sensor measurement with leap detection mechanism.

Secondly TS-0004[3] specifies procedures and serializations in independent manner from protocol bindings. Resource type and protocol data unit are defined using XSD for XML serialization. Mapping between XML and other serializations are also specified.

Thirdly TS-0008, TS-0009, TS-0010, TS-0020 specify protocol specific details for CoAP, HTTP, MQTT and Web Socket respectively.

2.3 Long name and short name

oneM2M introduced two types of notation, called long name and short name for resource types, attribute and so on. Long name is human friendly string and specifications mainly use this notation, while short name is short string consist of typically 2 or 3 characters (but not limited and sometimes longer) and communication protocol use this notation. In most cases, the initial characters of long name are assigned as short name, for examples, ct for CreationTime and at for AnnounceTo.

3 Problem Description

This section should be copied from the appropriate RFP(s). It is repeated here so it can be extended while the RFC authors learn more subtle details.

oneM2M specifies protocol based interface, but doesn't specify programing level API. As previously mentioned oneM2M allows variety of communication methods which are the combinations of 4 protocol bindings (HTTP, MQTT, CoAP, Websocket) and 3 serializations (XML, JSON, CBOR).

First problem is application portability. Without standardized API, application program tends to depend on the communication method initially intend to use and it will became hard to run another environment in which uses another communication method. (For example, an application designed for XML/HTTP, tend to run on environment use JSON/Websocket)

Second problem is the latency of the communication between CSE and application. Even if CSE and application is located in the same box, current oneM2M specifications define methods through protocols which requires serialization/deserialization of data, context-switch of applications, validation of incoming data and resulted in large latency compared to the situation both CSE and Application resides in the same Java VM and communicate with Java interfaces. Large latency reduces applicable area of oneM2M based solution.

Third problem is the complexity of handling of long name and short name. Even if short name is defined by trying to use initial characters, it is not straight forward to translate them in head.

4 Requirements

This section should be copied from the appropriate RFP(s)

- R0010 – The solution MUST provide means to access outer CSE from application.
- R0011 – The solution MUST provide means to access outer CSE from client CSE.
- R0012 – The solution MUST provide means to select a communication method for application.
- R0013 – The solution MUST provide means to select a communication method for client CSE.
- R0020 – The solution MUST provide means for CSE to accept requests form outer CSE.
- R0020 – The solution MUST provide means for CSE to accept requests form outer application.

- R0030 – The solution MUST provide means to communicate through Java interface between CSE and application that are located in the same OSGi framework.
- R0040 – The solution SHOULD hide differences of communication methods, which are combinations of 4 protocol bindings and 3 serializations (XML, JSON, CBOR).
- R0050 – The solution SHOULD provide developer friendly way for handling short names.
- R0060 – The solution MUST provide asynchronous interface using ‘call by value’, such as DTO.

5 Technical Solution

First give an architectural overview of the solution so the reader is gently introduced in the solution (Javadoc is not considered gently). What are the different modules? How do the modules relate? How do they interact? Where do they come from? This section should contain a class diagram. Then describe the different modules in detail. This should contain descriptions, Java code, UML class diagrams, state diagrams and interaction diagrams. This section should be sufficient to implement the solution assuming a skilled person.

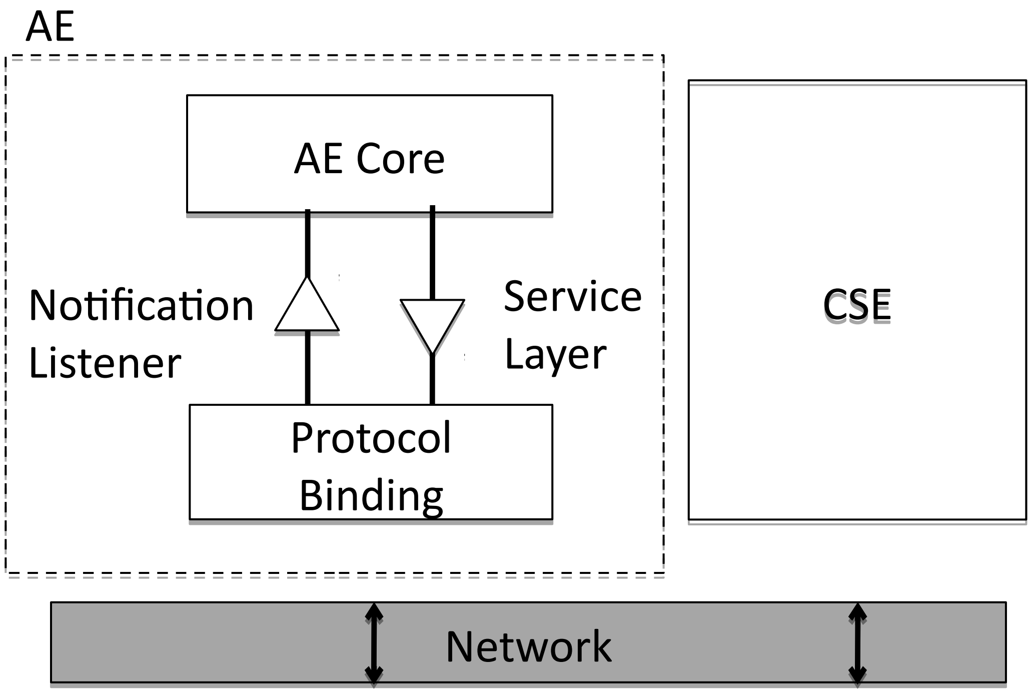
Strictly use the terminology a defined in the Problem Context.

On each level, list the limitations of the solutions and any rationales for design decisions. Almost every decision is a trade off so explain what those trade offs are and why a specific trade off is made.

Address what security mechanisms are implemented and how they should be used.

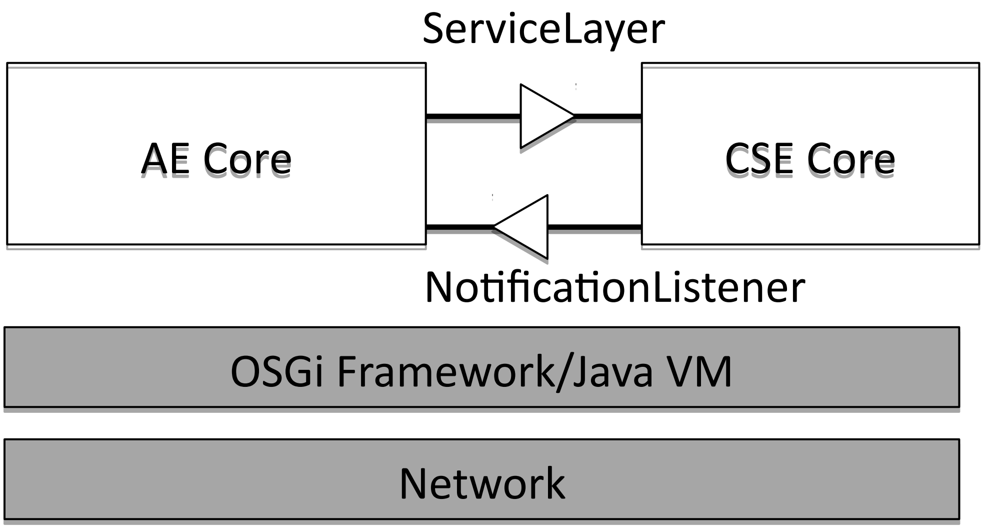
5.1 Overview for the solution

Protocol binding service is introduced to handle different protocols and serializations. oneM2M application uses the protocol binding service through Service Layer Interface to communicate CSE. The interface is protocol and serialization agnostic interface; it has no protocol and serialization specific methods, parameters, so that application can communicate to CSE without knowing which protocol is actually used.

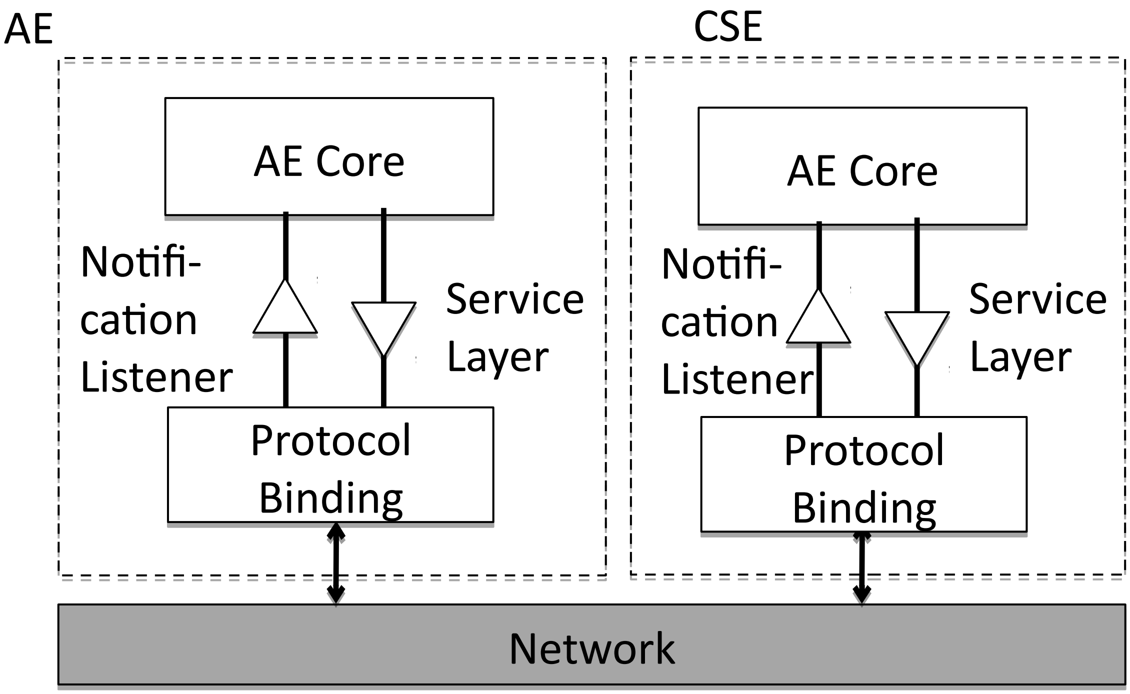


In figure, the term of Core is introduced for AE Core and CSE Core. This is for specifying parts which does not contain ProtocolBinding of AE and CSE.

Another use case is that the AE Core and the CSE core are located on the same OSGi Framework. In this use case, the AE Core and the CSE core communicate directly with ServiceLayer API, without inter-mediating ProtocolBinding Services. Following figure depicts overall configuration. Though this type of communication is not clearly defined in oneM2M specification, communicating directly without serializing data between AE and CSE allows shorter latency and less computational resources.



Following figure (right hand side) shows potential implementation of CSE, which are consisted fo CSE core and Protocol Binding Service, as symmetric to AE side. APIs defined in this RFC is consistent with the usage, but this RFC does not mandate that usage and it is left to implementor's choice.



5.2 Service Layer Interfaces

Service Layer Interface is for allowing AE to send request and get response.

request() method allows very raw data type access and it enables all possible message exchanges among oneM2M entities,

```
	Promise<ResponseDTO> request(RequestDTO request);
```

Meanwhile, it can be redundant to application developers, because they need to write composition of requestPrimitive and decomposition of responsePrimitive. This interface is provided for application developer allowing less application codes. It provides methods with higher level of abstraction; operation level of resource such as create, retrieve, update, delete and so on. They do not cover all of possible message exchange but do typical ones.

Note: If this RFC doesn't provide these methods, developers likely to create similar ones in their own (various) way.

```
package org.osgi.service.onem2m.servicelayer;
```

```
import org.osgi.onem2m.dto.RequestDTO;
import org.osgi.onem2m.dto.ResponseDTO;
import org.osgi.util.promise.Promise;

/**
 * Service Layer Interface, which locates between AE and Protocol Binding Service.
 */
public interface ServiceLayer {
        /**
         * send a request.
         *
         * @param request request
         * @return promise for ResponseDTO.
         */
        Promise<ResponseDTO> request(RequestDTO request);

        /**
         * create resource
         *
         * @param uri URI for parent resource
         * @param resource resource data
         * @return Promise of created resource
         */
        public Promise<ResourceDTO> create(String uri, ResourceDTO resource);

        /**
         * retrieve resource
         *
         * @param uri URI for retrieving resource
         * @return retrieved resource data
         */
        public Promise<ResourceDTO> retrieve(String uri, ResourceDTO resource);

        /**
         * retrieve subset of attributes.
         *
         * @param uri URI for retrieving resource
         * @param targetNames attribute names for retrival
         * @return retrieved resource data
         */
        public Promise<ResourceDTO> retrieve(String uri, List<String>
targetAttributes);

        /**
         * update resource
         *
         * @param uri URI for updating resource
         * @param resource data resource
```

```
     * @return updated resource
     */
    public Promise<ResourceDTO> update(String uri, ResourceDTO resource);

    /**
     * delete resource
     *
     * @param uri target URI for deleting resource
     */
    public Promise<Boolean> delete(String uri);

    /**
     * find resources
     *
     * @param uri URI for top of search
     * @param fc filter criteria
     * @return list of URIs matching the condition specified in fc
     */
    public Promise<List<String>> discovery(String uri, FilterCriteriaDTO fc);

        /**
     * find resources
     *
     * @param uri URI for top of search
     * @param fc filter criteria
     * @param drt Discovery Result Type (structured/unstructured)
     * @return list of URIs matching the condition specified in fc
     */
    public Promise<List<String>> discovery(String uri, FilterCriteriaDTO
fc, RequestPrimitiveDTO.DiscoveryResultType drt);

    /**
     * send notification
     *
     * @param notification
     */
    public Promise<Boolean> notify(String uri, NotificationDTO notification );

}
```

5.3 Service Property for Interfaces

~~Services implementing Service Layer Interface shall be registered with following properties.~~

Following table summarizes the service properties for services.

Interface	property Name	type	explanation
ServiceLayer	*No Property* ~~POAforAE~~	~~String[]~~	~~URIs for point of access. POA is basically set of listening protocol, host, port and subaddress or oneM2M address.~~ ~~In case that implementing entity is Protocol Binding, this property MUST be provided, meanwhile, in case of CSE Core, the property is MUST not provided.~~ ~~[MAY BE TRICKY]~~
NotificationListener	*No Property*~~AE-ID~~	~~String~~	~~ID of Application Entity~~
	~~POA~~	~~String[]~~	~~Point of Access~~

~~Here, AE-ID of NotificationListener is assigned by CSE thorough registration process, which is done by creating AE resource. After receiving assigned value, AE Core should updated the property with the value.~~

~~NOTE: It may be informative to put information on service properties, such as protocol, serialization, usage of secure protocol, oneM2M release version, or CSEtype, but there is no clear use case to use them, This RFC does not specify their usage.~~

5.4 Service Binding

Proper Protocol Binding Service must be bound to proper AE Core. Implementation of ServiceLayer may be registered as ServiceFactory on the service registry and implementation should create a proper service instance depending on calling AE Core.

How to create and configure is out of scope of this RFC.

5.5 Data Modification in Protocol Binding

As an only exception of the behavior of Protocol Binding entity (implements ServiceLayer), it MUST add pointOfAccess attribute using configured value, in following condition.

1. create() for <AE> resource

2. update() for <AE> only when ‘pointOfAccess’ is specified.

3. request() when content is equivalent to above.

This is because AE Core entity doesn’t know the pointOfAccess information and processing show above simply solve the problem.

5.6 Example: Registration

This section explains how application registers to its hosting CSE. In order to interact with the oneM2M system the bundle obtains a reference to the ServiceLayer service from the service registry

```
@Component
public class MyLightSwitchComponent {
@Reference
ServiceLayer client
}
```

After getting client, it starts registration by creating <AE> resource.<AE> resource creation requires App-ID, AE-ID, requestReachability attributes. In the following code, “C” is passed for AE-ID, this means asking CSE for assigning the value. Assigned value is is included in returned resource

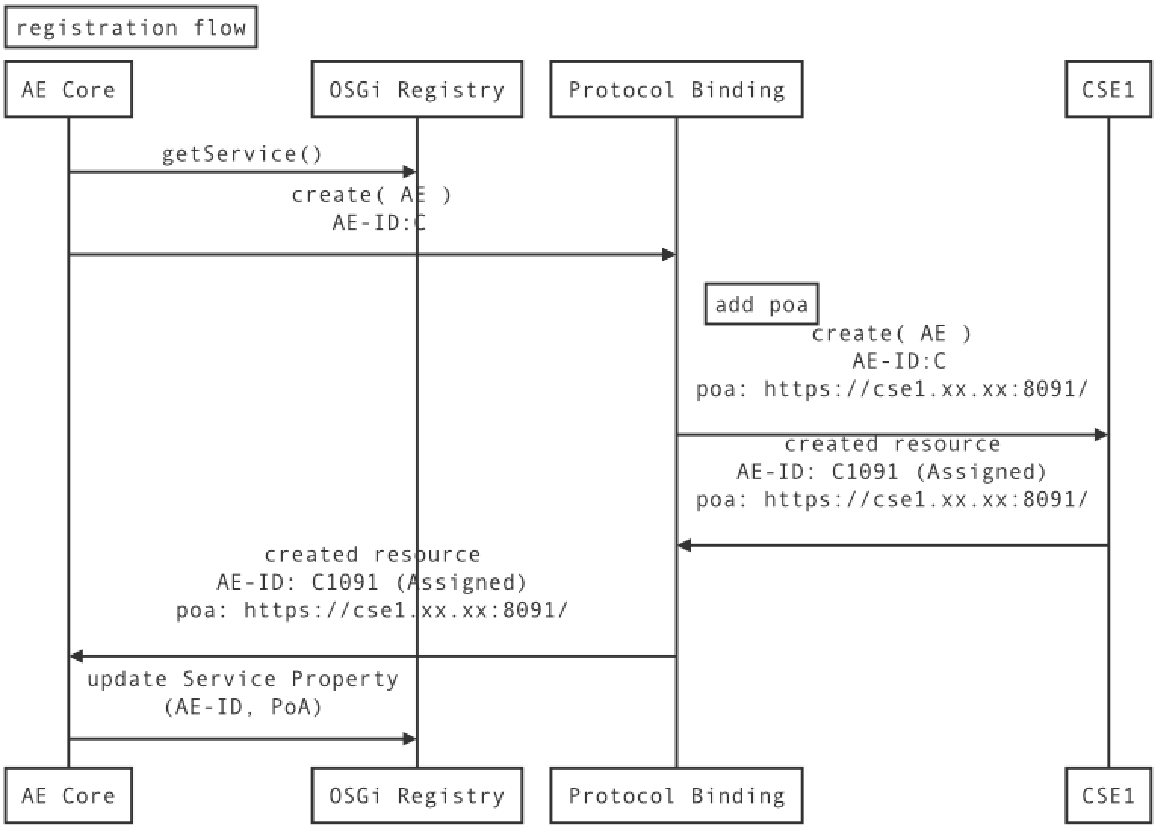
The information of pointOfAccess is kept in Protocol Binding entity, it is assigned by Protocol Binding entity. before sending the request message to network. The value is also returned in response.

After success response of <AE> resource, it registers NotificationListener with returned AE-ID and pointOfAccess as the service property

```
@Component
public class RegistrationExample implements NotificationListener {
      @Reference
      ServiceLayer client;

      BundleContext context;
       @Activate
       void start(BundleContext context) {
             this.context = context;

             // create AE: This means registration.
             ResourceDTO dto = new ResourceDTO();
             dto.resourceType = ResourceType.AE.getValue();
             dto.attribute.put("App-ID", "01.com.company.lightApp1<");
             dto.attribute.put("AE-ID", "C");
             dto.attribute.put("requestReachability", Boolean.TRUE);
             final Promise<ResourceDTO> ret = client.create("/CSE1/-", dto);

             ret.onResolve(new Runnable() {
                    public void run() {
                           ResourceDTO ae = null;
```

```
                          try {
                                ae = ret.getValue();
                          } catch (Exception e) {
                                e.printStackTrace();
                                return;
                          }
                           Dictionary prop = new Properties();
                           prop.put("AE-ID",ae.attribute.get("AE-ID"));
                           prop.put("PoA",ae.attribute.get("pointOfAccess");
                           context.registerService(new String[] {"org.osgi.service.onem2m.
NotificationListener"}, this, prop);
                    }
             });
      }

      @Override
      public void notified(RequestPrimitiveDTO request) {
            //....
      }
```

Ffollowing figure shows sequence diagram of the registration flow.



5.7 Example: Turning Light ON

This section explains how application turns on the lights located on remote site. There are mainly two ways to represent devices in oneM2M.

First way uses <flecContainer> resource type for representing device, its status, and so on. The resource type is introduced in oneM2M release 2 and it allows having custom attributes in it. Based on the resource, variety of the data model for devices, especially of home domain, are specified in TS-0023. With this way application can use standardized data model and operate device status in commonly used manner.

Second way uses <container> and <contentInstance> resource types for representing device and its status. These resource types are introduced very beginning of oneM2M and this approach is well explained in developer guide (TR-0025 [TODO ref. WEB]) Meanwhile, the resource type is not primarily designed for this purpose, but for storing data, so that how to operate device could be different from usual manner. To change state, new <contentInstance> is created with new status in its content attribute. <container> usually has multiple <contentInstance>s underneath and the latest one is supposed to be the latest status.

5.7.1 Example (Using <flexContainer>

In this example a simple bundle wishes to swtich light devices on when the bundle is started, and switch them off again when the bundle is stopped. Here it is assumed that registration process described in the previous section is done.

1. When the bundle has obtained a reference to the ServiceLayer then it can use the `discover` method to find all of the lightbulbs in the system

```
public class MyLightSwitchComponent {
@Reference
ServiceLayer client;

Promise<List<String>> discoveredLightbulbs;

@Activate
void start() {
  discoveredLightbulbs = findLightBulbs();
}

private Promise<List<String>> findLightBulbs() {
    String baseURI = "/homegateway/-/"; // - means CSE Base ( kaid of top directory )

    FilterCriteriaDTO filter = new FilterCriteriaDTO();
    filter.resourceType = Collections.singletonList(
      ResourceType.flexContainer.getValue());
    AttributeDTO attr = new AttributeDTO();
    attr.name = "contentDefinition";
    attr.value = "org.onem2m.home.device.light";
    filter.attribute = Collections.singletonList(attr);
    filter.filterOperation = FilterCriteriaDTO.FilterOperation.AND;

    return client.discovery(baseURI, filter);
}
}
```

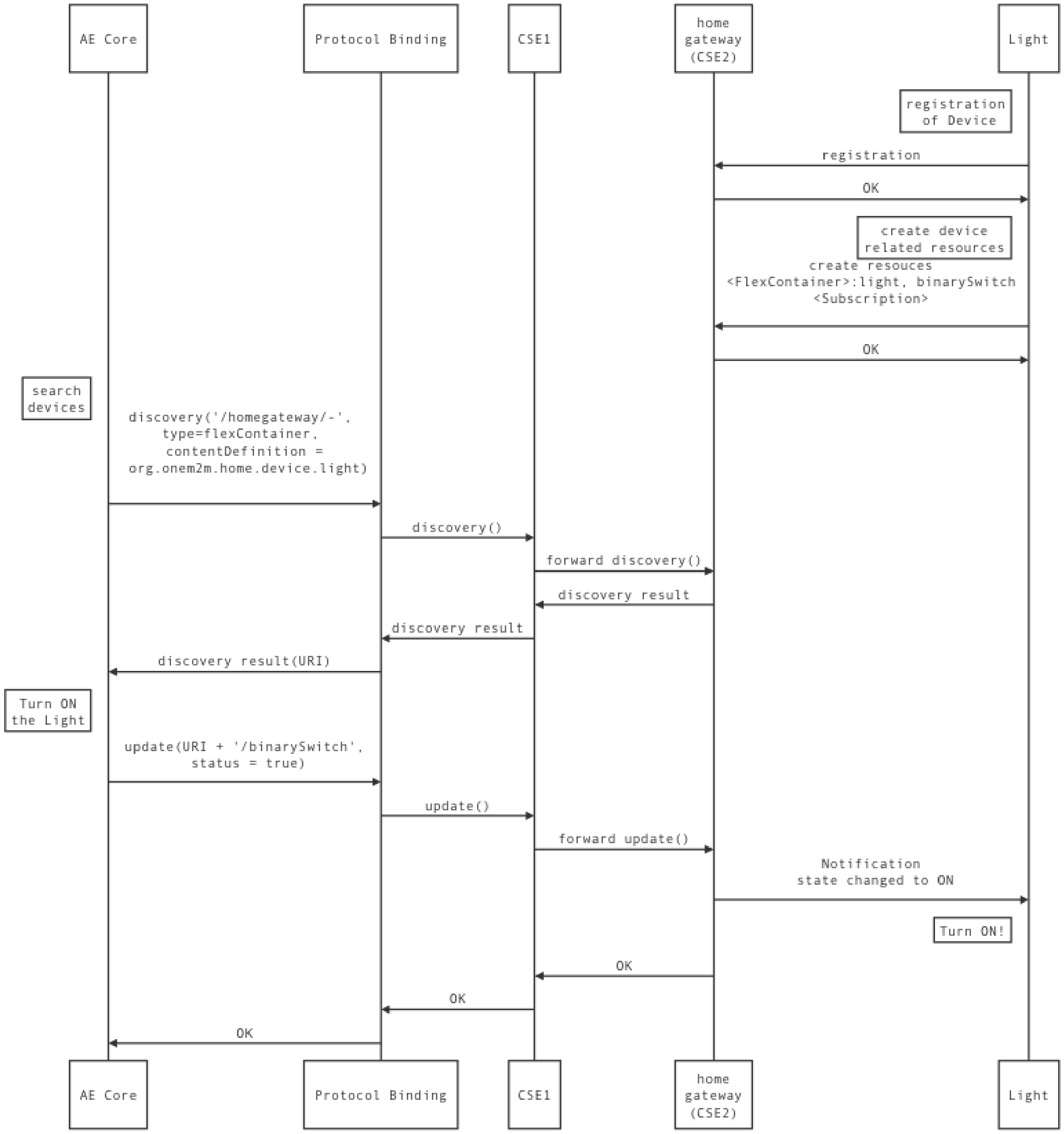
2. Once the lightbulbs are discovered then the bundle can switch on the bulbs by creating a content instance using the `update` method

```
public class MyLightSwitchComponent {
@Reference
ServiceLayer client;
Promise<List<ResourceDTO> allTurnedOn;

@Activate
void start() {
  allTurnedOn = findLightBulbs().
         flatMap(l -> Promises.all(
             l.stream()
                .map(this::turnOn)
                .collect(toList())));
}

private Promise<ResourceDTO> turnOn(String bulbUri) {
  ResourceDTO dto = new ResourceDTO();
  dto.attribute.put("powerStatus", Boolean.TRUE);
  return client.update(bulbUri + "/binarySwitch", dto );
}
}
```

3. Finally, the bulbs can be turned off again when the bundle is stopped.

```
public class MyLightSwitchComponent {
@Reference
ServiceLayer client;
Promise<List<ResourceDTO> allTurnedOn;

@Deactivate
void stop() {
    findLightBulbs().flatMap(l -> Promises.all(
        l.stream()
           .map(this::turnOff)
      .collect(toList())));
}
}
```

Following figure shows the example showing how application turn on the light device on the remote.



5.7.2 Example (Using <container> and <contentInstance>

In this example, device is expressed as <container> resource type in remote CSE (Home Gateway). Discovery is changed as follows. Here it assumes that all <container>s representing lightbulb have label of “lightBulb”.

```
private Promise<List<String>> findLightBulbs() {
```

```
         String baseURI = "/homegateway/-/"; // - means CSEBase ( kaid of top directory )

         FilterCriteriaDTO filter = new FilterCriteriaDTO();
         filter.resourceType = Collections.singletonList(
             ResourceType.container.getValue());
         filter.labels = Collections.singletonList("lightBulb");
         filter.filterOperation = FilterCriteriaDTO.FilterOperation.AND;
         return client.discovery(baseURI, filter);
}
```

Light can be controlled by creating new <contentInstance> resource as follows.

```
    private Promise<ResourceDTO> turnOn(String bulbUri) {
         ResourceDTO dto = new ResourceDTO();
         dto.resourceType = ResourceType.container.getValue();
         dto.attribute.put("content", "ON");
         return client.create(bulbUri , dto );
     }
```

6 Data Transfer Objects

RFC 185 defines Data Transfer Objects as a generic means for management solutions to interact with runtime entities in an OSGi Framework. DTOs provides a common, easily serializable representation of the technology.

For all new functionality added to the OSGi Framework the question should be asked: would this feature benefit from a DTO? The expectation is that in most cases it would.

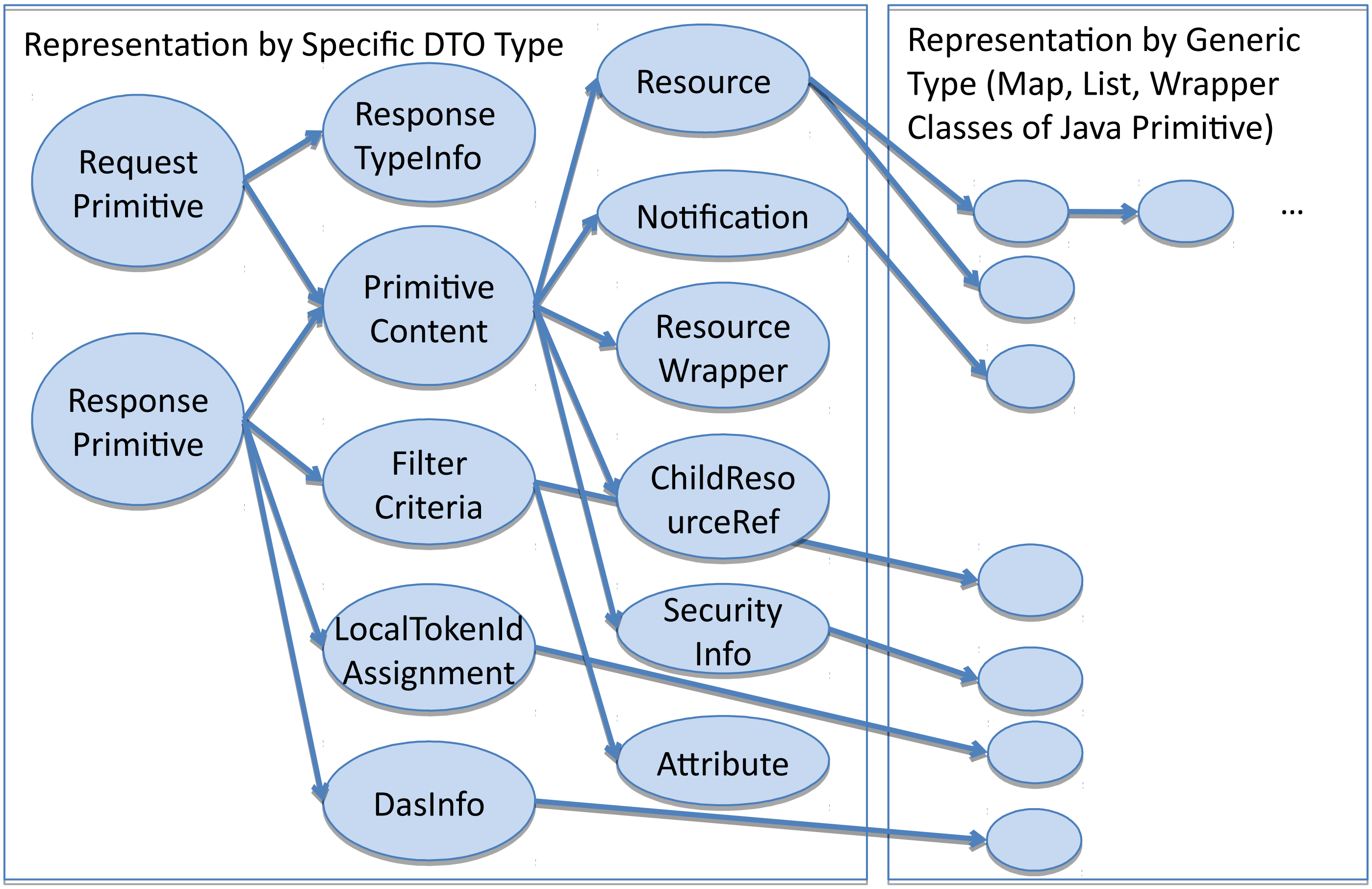
The DTOs for the design in this RFC should be described here and if there are no DTOs being defined an explanation should be given explaining why this is not applicable in this case.

This section is optional and could also be provided in a separate RFC.

6.1 Design Policy of DTOs

Data transfer Object was chosen as data object passing by the interfaces, following OSGi's design convention. Though data structure near root structures are designed specific DTO, deeper data types are to be stored as generic types, such as Map, List and Wrapper classes of Java Primitives. Following figure depict reference relationship among object. (See also the considered Alternatives)

In the class definition, some data types are shown as Object, but the assigned value shall be a type that is allowed for OSGi DTO.



In figure, followed 'DTO' is ommited.

6.2 RequestPrimitiveDTO

RequestPrimitiveDTO holds a Request Information used for oneM2M communication.

```
package org.osgi.service.onem2m.dto;
```

```
import java.util.*;

public class RequestPrimitiveDTO extends org.osgi.dto.DTO {
	@javax.xml.bind.annotation.XmlElement(required = true)
	public Operation operation;
	@javax.xml.bind.annotation.XmlElement(required = true)
	public String to;
	public String from;
	@javax.xml.bind.annotation.XmlElement(required = true)
	public String requestIdentifier;
	@javax.xml.bind.annotation.XmlElement(required = false)
	public Integer resourceType;
	public PrimitiveContentDTO content;
	public List<String> roleIDs;

	public String originatingTimestamp;
	public String requestExpirationTimestamp;
	public String resultExpirationTimestamp;
	public String operationExecutionTime;

	public ResponseTypeInfoDTO responseType;
	public String resultPersistence;
	@javax.xml.bind.annotation.XmlElement(required = false)
	public ResultContent resultContent;
	public String eventCategory;
	@javax.xml.bind.annotation.XmlElement(required = false)
	public Boolean deliveryAggregation;
	public String groupRequestIdentifier;
	public FilterCriteriaDTO filterCriteria;
	@javax.xml.bind.annotation.XmlElement(required = false)
	public DiscoveryResultType discoveryResultType;
	public String tokens;
	public List<String> tokenIDs;
	public List<String> localTokenIDs;
	@javax.xml.bind.annotation.XmlElement(required = false)
	public Boolean tokenReqIndicator;

	// Added at R3.0
	public List<String> groupRequestTargetMembers;
	public Boolean authSignatureIndicator;
	public List<String> authSignature;
	public Boolean authRelationshipIndicator;
	public Boolean semanticQueryIndicator;
	public ReleaseVersion releaseVersion;
	public String verndorInformation;
	public static enum DiscoveryResultType {
		structured(1), unstructured(2);
		// omitted
	}

	public static enum ResultContent {
		nothing(1), attributes(2), hierarchicalAddress(3),
		hierarchicalAddressAndAttributes(4),
attributesAndChildResources(5),
attributesAndChildResourceReferences(6),
childResourceReferences(7), originalResource(8), childResources(9);

		// omitted
	}

	public static enum Operation {
		Create(1), Retrieve(2), Update(3), Delete(4), Notify(5);
		// omitted
	}

}
```

6.3 ResponsePrimitiveDTO

ResponsePrimitiveDTO holds a Response Information used for oneM2M communication.

```
package org.osgi.service.onem2m.dto;
import java.util.*;

public class ResponsePrimitiveDTO extends org.osgi.dto.DTO{
	@javax.xml.bind.annotation.XmlElement( required  = true)
	public Integer responseStatusCode;
	@javax.xml.bind.annotation.XmlElement( required  = true)
	public String requestIdentifier;
	public PrimitiveContentDTO content;
	public String to;
	public String from;
	public String originatingTimestamp;
	public String resultExpirationTimestamp;
	public String eventCategory;
	@javax.xml.bind.annotation.XmlElement( required  = false)
	public ContentStatus contentStatus;
	@javax.xml.bind.annotation.XmlElement( required  = false)
	public Integer contentOffset;
	public List<LocalTokenIdAssignmentDTO> 
assignedTokenIdentifiers;//Map<String,Object>
	public List<DasInfoDTO> tokenReqInfo;//DynAuthTokenReqInfoDTO

	// Added R3.0
	public Boolean AuthSignatureReqInfo;
	public ReleaseVersion releaseVersionIndicator;
	public String vendorInformation;

	public static enum ContentStatus{
		PARTIAL_CONTENT, // 1
		FULL_CONTENT;  //2

	}
}
```

6.4 ResponseTypeInfoDTO

```
package org.osgi.service.onem2m.dto;

import java.util.*;

public class ResponseTypeInfoDTO extends org.osgi.dto.DTO {
	@javax.xml.bind.annotation.XmlElement(required = true)
	public java.lang.Integer responseTypeValue;
	@javax.xml.bind.annotation.XmlElement(required = true)
	public List<java.lang.String> notificationURI;

	public static enum ResponseType {
		nonBlockingRequestSynch(1), nonBlockingRequestAsynch(2), 
blockingRequest(3), flexBlocking(4);
		// omitted.
	}
}
```

6.5 FilterCriteriaDTO

```
package org.osgi.service.onem2m.dto;
import java.util.*;

public class FilterCriteriaDTO extends org.osgi.dto.DTO{
	public String createdBefore;
	public String createdAfter;
public String modifiedSince;
public String unmodifiedSince;
@javax.xml.bind.annotation.XmlElement( required  = false)
public Integer stateTagSmaller;
@javax.xml.bind.annotation.XmlElement( required  = false)
public Integer stateTagBigger;
public String expireBefore;
public String expireAfter;
public List<String> labels;
public List<Integer> resourceType;
@javax.xml.bind.annotation.XmlElement( required  = false)
public Integer sizeAbove;
@javax.xml.bind.annotation.XmlElement( required  = false)
public Integer sizeBelow;
public List<String> contentType;
public AttributeDTO attribute;
@javax.xml.bind.annotation.XmlElement( required  = false)
public FilterUsage filterUsage;
@javax.xml.bind.annotation.XmlElement( required  = false)
public Integer limit;
public String semanticsFilter;
@javax.xml.bind.annotation.XmlElement( required  = false)
public FilterOperation filterOperation;
@javax.xml.bind.annotation.XmlElement( required  = false)
public Integer contentFilterSyntax;
public String contentFilterQuery;
@javax.xml.bind.annotation.XmlElement( required  = false)
public Integer level;
@javax.xml.bind.annotation.XmlElement( required  = false)
public Integer offset;

// added in R3
public List<String> childLabels;
public List<String> parentLabels;
public String labelsQuery;
public Integer childResourceType;
public Integer parentResourceType;
public AttributeDTO childAttribute;
public AttributeDTO parentAttribute;
public String applyRelativePath;


public static enum FilterOperation {
      AND(1), OR(2);
      // omitted...
}
```

```
      public static enum FilterUsage {
            DiscoveryCriteria(1), ConditionalRetrival(2), IPEOndemandDiscovery(3);
            // omitted...

      }
}
```

6.6 ResourceDTO

```
package org.osgi.service.onem2m.dto;
import java.util.*;

public class ResourceDTO extends org.osgi.dto.DTO{

      // Universal Attribute, which can be held by all resources.
      @javax.xml.bind.annotation.XmlElement( required  = true)
      public Integer resourceType;
      @javax.xml.bind.annotation.XmlElement( required  = true)
      public String resourceID;
      @javax.xml.bind.annotation.XmlElement( required  = true)
      public String parentID;
      @javax.xml.bind.annotation.XmlElement( required  = true)
      public String creationTime;
      @javax.xml.bind.annotation.XmlElement( required  = true)
      public String lastModifiedTime;

      public String resourceName;

      // optional, Universal Attributes
      public List<String> labels;

      /**
       * Non Universal Attribute.
       * Value Part must be the types that are allowed for OSGi DTO.
       */
      public Map<String, Object> attribute;
}
```

6.7 NotificationDTO

NotificationDTO has information of notification.

```java
package org.osgi.service.onem2m.dto;
import java.util.*;

public class NotificationDTO extends org.osgi.dto.DTO{
      public Map<String,Object> notificationEvent;//NotificationEventDTO
      @javax.xml.bind.annotation.XmlElement( required  = false)
      public Boolean verificationRequest;
      @javax.xml.bind.annotation.XmlElement( required  = false)
      public Boolean subscriptionDeletion;
      public String subscriptionReference;
      public String creator;
      public String notificationForwardingURI;
      @javax.xml.bind.annotation.XmlElement( required  = false)
      public Map<String,Object> ipeDiscoveryRequest;//IPEDiscoveryRequestDTO
}
```

6.8 Other DTOs

There are some other DTOs, please refer Javadoc section for them.

6.9 Mapping Rules for Generic DTO

Following table summarizes mapping rule between oneM2M data types and Generic types used in DTOs. There are two types of XSD are defined in oneM2M, which are longname version and shortname version. The longname version should be refered.

oneM2M Types (XML Schema)	Type of OSGi DTO	
Basic Types of XML Schema	Wrapper Object of Java primitive	For example: xs:integer, xs:float
xs:anyURI, m2m:ID,	String	
m2m:timestamp	String	`YYYYMMDDThhmmss,ssssss`
m2m:absRelTimestamp	String	Union of m2m:timestamp and xs:long. This is exception of union rule above. Distinction is done by existence of ‘T’
xs:sequence (as complexType)	Map	Name of element is used for key of map.
xs:list, xs:sequence (as list)	List	
xs:union	Map	Base attribute of restriction tag is used for key of map. Only one key is allowed. See Example of missingDataList:

Following XML is an example of missingData.

```
<xs:simpleType name="missingDataList">
      <xs:union>
        <simpleType>
            <restriction base='m2m:listOfTimeStamp' />
        </simpleType>
        <simpleType>
            <restriction base='m2m:listOfRelTimeStamp' />
        </simpleType>
      </xs:union>
    </xs:simpleType>
```

7 Javadoc

Please include Javadoc of any new APIs here, once the design has matured. Instructions on how to export Javadoc for inclusion in the RFC can be found here: https://www.osgi.org/members/RFC/Javadoc

Demo Documentation

18/06/29 14:26

Package Summary		Page
org.osgi.service.onem2m	This package includes essential interfaces for oneM2M service layer API.	31
org.osgi.service.onem2m.dto	This package contains OSGi DTOs used in oneM2M Service Layer API.	37

Package org.osgi.service.onem2m

This package includes essential interfaces for oneM2M service layer API.

See:

Description

Interface Summary		*Page*
NotificationListener	Primary Interface for an oneM2M entity to send request and get response to/from other oneM2M entity.	33
ServiceLayer	Primary Interface for an oneM2M entity to send request and get response to/from other oneM2M entity.	34

Package org.osgi.service.onem2m Description

This package includes essential interfaces for oneM2M service layer API.

Interface NotificationListener

org.osgi.service.onem2m

`public interface` **`NotificationListener`**

Primary Interface for an oneM2M entity to send request and get response to/from other oneM2M entity.

Method Summary		*Page*
`void`	**`notified`**`(RequestPrimitiveDTO request)` receive notification.	33

Method Detail

notified

`void` **`notified`**`(RequestPrimitiveDTO request)`

receive notification.

Parameters:

`request` - request primitive

Interface ServiceLayer

org.osgi.service.onem2m

```
public interface ServiceLayer
```

Primary Interface for an oneM2M entity to send request and get response to/from other oneM2M entity.

Method Summary		Page
org.osgi.util.promise.Promise<ResourceDTO>	`create(String uri, ResourceDTO resource)` create resource	35
org.osgi.util.promise.Promise<Boolean>	`delete(String uri)` delete resource	36
org.osgi.util.promise.Promise<List<String>>	`discovery(String uri, FilterCriteriaDTO fc)` find resources.	36
org.osgi.util.promise.Promise<List<String>>	`discovery(String uri, FilterCriteriaDTO fc, RequestPrimitiveDTO.DiscoveryResultType drt)` find resources	36
org.osgi.util.promise.Promise<Boolean>	`notify(String uri, NotificationDTO notification)` send notification	36
org.osgi.util.promise.Promise<ResponsePrimitiveDTO>	`request(RequestPrimitiveDTO request)` send a request.	34
org.osgi.util.promise.Promise<ResourceDTO>	`retrieve(String uri, List<String> targetAttributes)` retrieve subset of attributes.	35
org.osgi.util.promise.Promise<ResourceDTO>	`retrieve(String uri, ResourceDTO resource)` retrieve resource	35
org.osgi.util.promise.Promise<ResourceDTO>	`update(String uri, ResourceDTO resource)` update resource	35

Method Detail

request

```
org.osgi.util.promise.Promise<ResponsePrimitiveDTO> request(RequestPrimitiveDTO request)
```

send a request.

Parameters:

`request` - request primitive

Returns:

promise of ResponseDTO.

create

```
org.osgi.util.promise.Promise<ResourceDTO> create(String uri,
                                                  ResourceDTO resource)
```

create resource

Parameters:

`uri` - URI for parent resource

`resource` - resource data

Returns:

Promise of created resource

retrieve

```
org.osgi.util.promise.Promise<ResourceDTO> retrieve(String uri,
                                                    ResourceDTO resource)
```

retrieve resource

Parameters:

`uri` - URI for retrieving resource

Returns:

retrieved resource data

retrieve

```
org.osgi.util.promise.Promise<ResourceDTO> retrieve(String uri,
                                                    List<String> targetAttributes)
```

retrieve subset of attributes.

Parameters:

`uri` - URI for retrieving resource

`targetAttributes` - names of the target attribute

Returns:

retrieved resource data

update

```
org.osgi.util.promise.Promise<ResourceDTO> update(String uri,
                                                  ResourceDTO resource)
```

update resource

Parameters:

`uri` - URI for updating resource

`resource` - data resource

Returns:

updated resource

delete

```
org.osgi.util.promise.Promise<Boolean> delete(String uri)
```

delete resource

Parameters:

`uri` - target URI for deleting resource

discovery

```
org.osgi.util.promise.Promise<List<String>> discovery(String uri,
                                                      FilterCriteriaDTO fc)
```

find resources. Discovery Result Type is kept as blank and default value of target CSE is used for the parameter.

Parameters:

`uri` - URI for top of search

`fc` - filter criteria

Returns:

list of URIs matching the condition specified in fc

discovery

```
org.osgi.util.promise.Promise<List<String>> discovery(String uri,
                                                      FilterCriteriaDTO fc,
                                                      RequestPrimitiveDTO.DiscoveryResultType
drt)
```

find resources

Parameters:

`uri` - URI for top of search

`fc` - filter criteria

`drt` - Discovery Result Type (structured/unstructured)

Returns:

list of URIs matching the condition specified in fc

notify

```
org.osgi.util.promise.Promise<Boolean> notify(String uri,
                                              NotificationDTO notification)
```

send notification

Package org.osgi.service.onem2m.dto

This package contains OSGi DTOs used in oneM2M Service Layer API.

See:

Description

Class Summary		***Page***
AttributeDTO	DTO expresses Attribute.	39
ChildResourceRefDTO	DTO expressing ChildResourceRef.	40
DasInfoDTO	DTO expressing DasInfo.	42
FilterCriteriaDTO	DTO expressing FilterCriteria.	43
LocalTokenIdAssignmentDTO	DTO expressing LocalTokenIdAssignment.	52
NotificationDTO	DTO expressing Notification.	53
PrimitiveContentDTO	DTO expressing Primitive Content.	56
RequestPrimitiveDTO	DTO expresses Request Primitive.	59
ResourceDTO	DTO expressing Resource.	70
ResourceWrapperDTO	DTO expressing ResourceWrapper.	80
ResponsePrimitiveDTO	DTO expressing Response Primitive.	81
ResponseTypeInfoDTO	Expressing ResponseTypeInfo	86
SecurityInfoDTO	DTO expressing Security Info.	89

Enum Summary		***Page***
FilterCriteriaDTO.FilterOperation		48
FilterCriteriaDTO.FilterUsage		50
ReleaseVersion	Enum expressing oneM2M specification version.	57
RequestPrimitiveDTO.DiscoveryResultType		64
RequestPrimitiveDTO.Operation		66

RequestPrimitiveDTO.ResultContent		68
ResourceType		72
ResponsePrimitiveDTO.ContentStatus		84
ResponseTypeInfoDTO.ResponseType		87

Exception Summary		*Page*
OneM2MException	General Exception for oneM2M.	55

Package org.osgi.service.onem2m.dto Description

This package contains OSGi DTOs used in oneM2M Service Layer API.

Class AttributeDTO

org.osgi.service.onem2m.dto

```
java.lang.Object
 └org.osgi.dto.DTO
   └org.osgi.service.onem2m.dto.AttributeDTO
```

```
public class AttributeDTO
extends org.osgi.dto.DTO
```

DTO expresses Attribute. This is typically used in FilterCriteriaDTO

Field Summary		*Page*
String	**name**	39
Object	**value**	39

Constructor Summary	*Page*
AttributeDTO()	39

Methods inherited from class org.osgi.dto.DTO
toString

Field Detail

name

```
public String name
```

value

```
public Object value
```

Constructor Detail

AttributeDTO

```
public AttributeDTO()
```

Class ChildResourceRefDTO

org.osgi.service.onem2m.dto

```
java.lang.Object
 └org.osgi.dto.DTO
   └org.osgi.service.onem2m.dto.ChildResourceRefDTO
```

```
public class ChildResourceRefDTO
extends org.osgi.dto.DTO
```

DTO expressing ChildResourceRef.

Field Summary		*Page*
String	name	40
String	specializationID	40
Integer	type	40
String	uri	40

Constructor Summary	*Page*
ChildResourceRefDTO()	41

Methods inherited from class org.osgi.dto.DTO
toString

Field Detail

uri

`public String uri`

name

`public String name`

type

`public Integer type`

specializationID

`public String specializationID`

Constructor Detail

ChildResourceRefDTO

```
public ChildResourceRefDTO()
```

Class DasInfoDTO

org.osgi.service.onem2m.dto

```
java.lang.Object
 └org.osgi.dto.DTO
   └org.osgi.service.onem2m.dto.DasInfoDTO
```

```
public class DasInfoDTO
extends org.osgi.dto.DTO
```

DTO expressing DasInfo. DAS is short for Dynamic Authorization Server.

Field Summary		Page
Map<String,Object>	dasRequest	42
String	securedDasRequest	42
String	uri	42

Constructor Summary	Page
DasInfoDTO()	42

Methods inherited from class org.osgi.dto.DTO
toString

Field Detail

uri

```
public String uri
```

dasRequest

```
public Map<String,Object> dasRequest
```

securedDasRequest

```
public String securedDasRequest
```

Constructor Detail

DasInfoDTO

```
public DasInfoDTO()
```

Class FilterCriteriaDTO

org.osgi.service.onem2m.dto

```
java.lang.Object
 └org.osgi.dto.DTO
   └org.osgi.service.onem2m.dto.FilterCriteriaDTO
```

```
public class FilterCriteriaDTO
extends org.osgi.dto.DTO
```

DTO expressing FilterCriteria. This data structure is used for searching resources.

Nested Class Summary		Page
static enum	**FilterCriteriaDTO.FilterOperation**	48
static enum	**FilterCriteriaDTO.FilterUsage**	50

Field Summary		Page
String	**applyRelativePath**	47
List<AttributeDTO>	**attribute**	45
List<AttributeDTO>	**childAttribute**	47
List<String>	**childLabels**	46
List<Integer>	**childResourceType**	46
String	**contentFilterQuery**	46
Integer	**contentFilterSyntax**	46
List<String>	**contentType**	45
String	**createdAfter**	44
String	**createdBefore**	44
String	**expireAfter**	45
String	**expireBefore**	45
FilterCriteriaDTO.FilterOperation	**filterOperation**	46
FilterCriteriaDTO.FilterUsage	**filterUsage**	45
List<String>	**labels**	45
String	**labelsQuery**	46
Integer	**level**	46
Integer	**limit**	45
String	**modifiedSince**	44
Integer	**offset**	46
List<AttributeDTO>	**parentAttribute**	47
List<String>	**parentLabels**	46

List<Integer>	parentResourceType	47
List<Integer>	resourceType	45
List<String>	semanticsFilter	46
Integer	sizeAbove	45
Integer	sizeBelow	45
Integer	stateTagBigger	44
Integer	stateTagSmaller	44
String	unmodifiedSince	44

Constructor Summary	Page
FilterCriteriaDTO()	47

Methods inherited from class org.osgi.dto.DTO
toString

Field Detail

createdBefore

public String **createdBefore**

createdAfter

public String **createdAfter**

modifiedSince

public String **modifiedSince**

unmodifiedSince

public String **unmodifiedSince**

stateTagSmaller

public Integer **stateTagSmaller**

stateTagBigger

public Integer **stateTagBigger**

expireBefore

`public String expireBefore`

expireAfter

`public String expireAfter`

labels

`public List<String> labels`

resourceType

`public List<Integer> resourceType`

sizeAbove

`public Integer sizeAbove`

sizeBelow

`public Integer sizeBelow`

contentType

`public List<String> contentType`

attribute

`public List<AttributeDTO> attribute`

filterUsage

`public FilterCriteriaDTO.FilterUsage filterUsage`

limit

`public Integer limit`

semanticsFilter

```
public List<String> semanticsFilter
```

filterOperation

```
public FilterCriteriaDTO.FilterOperation filterOperation
```

contentFilterSyntax

```
public Integer contentFilterSyntax
```

contentFilterQuery

```
public String contentFilterQuery
```

level

```
public Integer level
```

offset

```
public Integer offset
```

childLabels

```
public List<String> childLabels
```

parentLabels

```
public List<String> parentLabels
```

labelsQuery

```
public String labelsQuery
```

childResourceType

```
public List<Integer> childResourceType
```

parentResourceType

```
public List<Integer> parentResourceType
```

childAttribute

```
public List<AttributeDTO> childAttribute
```

parentAttribute

```
public List<AttributeDTO> parentAttribute
```

applyRelativePath

```
public String applyRelativePath
```

Constructor Detail

FilterCriteriaDTO

```
public FilterCriteriaDTO()
```

Enum FilterCriteriaDTO.FilterOperation

org.osgi.service.onem2m.dto

```
java.lang.Object
 └java.lang.Enum<FilterCriteriaDTO.FilterOperation>
   └org.osgi.service.onem2m.dto.FilterCriteriaDTO.FilterOperation
```

All Implemented Interfaces:

Comparable<FilterCriteriaDTO.FilterOperation>, Serializable

Enclosing class:

FilterCriteriaDTO

```
public static enum FilterCriteriaDTO.FilterOperation
extends Enum<FilterCriteriaDTO.FilterOperation>
```

Enum Constant Summary	*Page*
AND	48
OR	48

Method Summary		*Page*
int	**getValue**()	49
static FilterCriteriaDTO.FilterOperation	**valueOf**(String name)	49
static FilterCriteriaDTO.FilterOperation[]	**values**()	48

Enum Constant Detail

AND

```
public static final FilterCriteriaDTO.FilterOperation AND
```

OR

```
public static final FilterCriteriaDTO.FilterOperation OR
```

Method Detail

values

```
public static FilterCriteriaDTO.FilterOperation[] values()
```

valueOf

```
public static FilterCriteriaDTO.FilterOperation valueOf(String name)
```

getValue

```
public int getValue()
```

Enum FilterCriteriaDTO.FilterUsage

org.osgi.service.onem2m.dto

```
java.lang.Object
 └java.lang.Enum<FilterCriteriaDTO.FilterUsage>
   └org.osgi.service.onem2m.dto.FilterCriteriaDTO.FilterUsage
```

All Implemented Interfaces:

Comparable<FilterCriteriaDTO.FilterUsage>, Serializable

Enclosing class:

FilterCriteriaDTO

```
public static enum FilterCriteriaDTO.FilterUsage
extends Enum<FilterCriteriaDTO.FilterUsage>
```

Enum Constant Summary	Page
ConditionalRetrival	50
DiscoveryCriteria	50
IPEOndemandDiscovery	50

Method Summary		Page
int	**getValue**()	51
static FilterCriteriaDTO.FilterUsage	**valueOf**(String name)	51
static FilterCriteriaDTO.FilterUsage[]	**values**()	51

Enum Constant Detail

DiscoveryCriteria

`public static final FilterCriteriaDTO.FilterUsage DiscoveryCriteria`

ConditionalRetrival

`public static final FilterCriteriaDTO.FilterUsage ConditionalRetrival`

IPEOndemandDiscovery

`public static final FilterCriteriaDTO.FilterUsage IPEOndemandDiscovery`

Method Detail

values

```
public static FilterCriteriaDTO.FilterUsage[] values()
```

valueOf

```
public static FilterCriteriaDTO.FilterUsage valueOf(String name)
```

getValue

```
public int getValue()
```

Class LocalTokenIdAssignmentDTO

org.osgi.service.onem2m.dto

```
java.lang.Object
 └org.osgi.dto.DTO
   └org.osgi.service.onem2m.dto.LocalTokenIdAssignmentDTO
```

```
public class LocalTokenIdAssignmentDTO
extends org.osgi.dto.DTO
```

DTO expressing LocalTokenIdAssignment.

Field Summary		Page
String	localTokenID	52
String	tokenID	52

Constructor Summary	Page
LocalTokenIdAssignmentDTO()	52

Methods inherited from class org.osgi.dto.DTO
toString

Field Detail

localTokenID

```
public String localTokenID
```

tokenID

```
public String tokenID
```

Constructor Detail

LocalTokenIdAssignmentDTO

```
public LocalTokenIdAssignmentDTO()
```

Class NotificationDTO

[org.osgi.service.onem2m.dto](#)

```
java.lang.Object
 └org.osgi.dto.DTO
   └org.osgi.service.onem2m.dto.NotificationDTO
```

```
public class NotificationDTO
extends org.osgi.dto.DTO
```

DTO expressing Notification.

Field Summary		*Page*
String	**creator**	54
Map<String ,Object>	**ipeDiscoveryRequest**	54
Map<String ,Object>	**notificationEvent**	53
String	**notificationForwardingURI**	54
Boolean	**subscriptionDeletion**	53
String	**subscriptionReference**	54
Boolean	**verificationRequest**	53

Constructor Summary	*Page*
NotificationDTO()	54

Methods inherited from class org.osgi.dto.DTO
toString

Field Detail

notificationEvent

```
public Map<String,Object> notificationEvent
```

verificationRequest

```
public Boolean verificationRequest
```

subscriptionDeletion

```
public Boolean subscriptionDeletion
```

subscriptionReference

```
public String subscriptionReference
```

creator

```
public String creator
```

notificationForwardingURI

```
public String notificationForwardingURI
```

ipeDiscoveryRequest

```
public Map<String,Object> ipeDiscoveryRequest
```

Constructor Detail

NotificationDTO

```
public NotificationDTO()
```

Class OneM2MException

org.osgi.service.onem2m.dto

```
java.lang.Object
 └java.lang.Throwable
   └java.lang.Exception
     └java.io.IOException
       └org.osgi.service.onem2m.dto.OneM2MException
```

All Implemented Interfaces:

Serializable

```
public class OneM2MException
extends IOException
```

General Exception for oneM2M.

Field Summary		Page
String	cause	55
int	errorCode	55

Constructor Summary	Page
OneM2MException()	55

Field Detail

errorCode

`public int errorCode`

cause

`public String cause`

Constructor Detail

OneM2MException

`public OneM2MException()`

Class PrimitiveContentDTO

org.osgi.service.onem2m.dto

```
java.lang.Object
 └org.osgi.dto.DTO
   └org.osgi.service.onem2m.dto.PrimitiveContentDTO
```

```
public class PrimitiveContentDTO
extends org.osgi.dto.DTO
```

DTO expressing Primitive Content. This Data structure is like union. Only one field MUST have a value, other field MUST be null.

Constructor Summary	Page
`PrimitiveContentDTO()`	56

Methods inherited from class org.osgi.dto.DTO
`toString`

Constructor Detail

PrimitiveContentDTO

```
public PrimitiveContentDTO()
```

Enum ReleaseVersion

org.osgi.service.onem2m.dto

```
java.lang.Object
 └java.lang.Enum<ReleaseVersion>
   └org.osgi.service.onem2m.dto.ReleaseVersion
```

All Implemented Interfaces:

Comparable<ReleaseVersion>, Serializable

```
public enum ReleaseVersion
extends Enum<ReleaseVersion>
```

Enum expressing oneM2M specification version. This information is introduced after Release 2.0 and oneM2M uses only R2A,R3_0 (as 2a and 3).

Enum Constant Summary	Page
R1_0	57
R1_1	57
R2_0	57
R2A	58
R3_0	58

Method Summary		Page
static ReleaseVersion	valueOf(String name)	58
static ReleaseVersion[]	values()	58

Enum Constant Detail

R1_0

```
public static final ReleaseVersion R1_0
```

R1_1

```
public static final ReleaseVersion R1_1
```

R2_0

```
public static final ReleaseVersion R2_0
```

R2A

```
public static final ReleaseVersion R2A
```

R3_0

```
public static final ReleaseVersion R3_0
```

Method Detail

values

```
public static ReleaseVersion[] values()
```

valueOf

```
public static ReleaseVersion valueOf(String name)
```

Class RequestPrimitiveDTO

org.osgi.service.onem2m.dto

```
java.lang.Object
 └org.osgi.dto.DTO
   └org.osgi.service.onem2m.dto.RequestPrimitiveDTO
```

```
public class RequestPrimitiveDTO
extends org.osgi.dto.DTO
```

DTO expresses Request Primitive.

Nested Class Summary

		Page
static enum	**RequestPrimitiveDTO.DiscoveryResultType**	64
static enum	**RequestPrimitiveDTO.Operation**	66
static enum	**RequestPrimitiveDTO.ResultContent**	68

Field Summary

		Page
Boolean	**authRelationshipIndicator**	63
List<String>	**authSignature**	63
Boolean	**authSignatureIndicator**	62
PrimitiveContentDTO	**content**	61
Boolean	**deliveryAggregation**	62
RequestPrimitiveDTO.DiscoveryResultType	**discoveryResultType**	62
String	**eventCategory**	61
FilterCriteriaDTO	**filterCriteria**	62
String	**from**	60
String	**groupRequestIdentifier**	62
List<String>	**groupRequestTargetMembers**	62
List<String>	**localTokenIDs**	62
RequestPrimitiveDTO.Operation	**operation**	60
String	**operationExecutionTime**	61
String	**originatingTimestamp**	61
ReleaseVersion	**releaseVersion**	63
String	**requestExpirationTimestamp**	61
String	**requestIdentifier**	60
Integer	**resourceType**	60
ResponseTypeInfoDTO	**responseType**	61

RequestPrimitiveDTO.ResultContent	**resultContent**	61
String	**resultExpirationTimestamp**	61
String	**resultPersistence**	61
List<String>	**roleIDs**	61
Boolean	**semanticQueryIndicator**	63
String	**to**	60
List<String>	**tokenIDs**	62
Boolean	**tokenReqIndicator**	62
String	**tokens**	62
String	**verndorInformation**	63

Constructor Summary	*Page*
RequestPrimitiveDTO()	63

Methods inherited from class org.osgi.dto.DTO
toString

Field Detail

operation

public RequestPrimitiveDTO.Operation **operation**

to

public String **to**

from

public String **from**

requestIdentifier

public String **requestIdentifier**

resourceType

public Integer **resourceType**

content

```
public PrimitiveContentDTO content
```

roleIDs

```
public List<String> roleIDs
```

originatingTimestamp

```
public String originatingTimestamp
```

requestExpirationTimestamp

```
public String requestExpirationTimestamp
```

resultExpirationTimestamp

```
public String resultExpirationTimestamp
```

operationExecutionTime

```
public String operationExecutionTime
```

responseType

```
public ResponseTypeInfoDTO responseType
```

resultPersistence

```
public String resultPersistence
```

resultContent

```
public RequestPrimitiveDTO.ResultContent resultContent
```

eventCategory

```
public String eventCategory
```

deliveryAggregation

```
public Boolean deliveryAggregation
```

groupRequestIdentifier

```
public String groupRequestIdentifier
```

filterCriteria

```
public FilterCriteriaDTO filterCriteria
```

discoveryResultType

```
public RequestPrimitiveDTO.DiscoveryResultType discoveryResultType
```

tokens

```
public String tokens
```

tokenIDs

```
public List<String> tokenIDs
```

localTokenIDs

```
public List<String> localTokenIDs
```

tokenReqIndicator

```
public Boolean tokenReqIndicator
```

groupRequestTargetMembers

```
public List<String> groupRequestTargetMembers
```

authSignatureIndicator

```
public Boolean authSignatureIndicator
```

authSignature

```
public List<String> authSignature
```

authRelationshipIndicator

```
public Boolean authRelationshipIndicator
```

semanticQueryIndicator

```
public Boolean semanticQueryIndicator
```

releaseVersion

```
public ReleaseVersion releaseVersion
```

verndorInformation

```
public String verndorInformation
```

Constructor Detail

RequestPrimitiveDTO

```
public RequestPrimitiveDTO()
```

Enum RequestPrimitiveDTO.DiscoveryResultType

org.osgi.service.onem2m.dto

```
java.lang.Object
 └java.lang.Enum<RequestPrimitiveDTO.DiscoveryResultType>
   └org.osgi.service.onem2m.dto.RequestPrimitiveDTO.DiscoveryResultType
```

All Implemented Interfaces:

Comparable<RequestPrimitiveDTO.DiscoveryResultType>, Serializable

Enclosing class:

RequestPrimitiveDTO

```
public static enum RequestPrimitiveDTO.DiscoveryResultType
extends Enum<RequestPrimitiveDTO.DiscoveryResultType>
```

Enum Constant Summary	*Page*
structured	64
unstructured	64

Method Summary		*Page*
int	**getValue**()	65
static RequestPrimitiveDTO.DiscoveryResultType	**valueOf**(String name)	65
static RequestPrimitiveDTO.DiscoveryResultType[]	**values**()	64

Enum Constant Detail

structured

`public static final RequestPrimitiveDTO.DiscoveryResultType structured`

unstructured

`public static final RequestPrimitiveDTO.DiscoveryResultType unstructured`

Method Detail

values

`public static RequestPrimitiveDTO.DiscoveryResultType[] values()`

valueOf

```
public static RequestPrimitiveDTO.DiscoveryResultType valueOf(String name)
```

getValue

```
public int getValue()
```

Enum RequestPrimitiveDTO.Operation

org.osgi.service.onem2m.dto

```
java.lang.Object
 └java.lang.Enum<RequestPrimitiveDTO.Operation>
   └org.osgi.service.onem2m.dto.RequestPrimitiveDTO.Operation
```

All Implemented Interfaces:

Comparable<RequestPrimitiveDTO.Operation>, Serializable

Enclosing class:

RequestPrimitiveDTO

```
public static enum RequestPrimitiveDTO.Operation
extends Enum<RequestPrimitiveDTO.Operation>
```

Enum Constant Summary	Page
Create	66
Delete	67
Notify	67
Retrieve	66
Update	66

Method Summary		Page
int	**getValue**()	67
static RequestPrimitiveDTO.Operation	**valueOf**(String name)	67
static RequestPrimitiveDTO.Operation[]	**values**()	67

Enum Constant Detail

Create

`public static final RequestPrimitiveDTO.Operation **Create**`

Retrieve

`public static final RequestPrimitiveDTO.Operation **Retrieve**`

Update

`public static final RequestPrimitiveDTO.Operation **Update**`

Delete

```
public static final RequestPrimitiveDTO.Operation Delete
```

Notify

```
public static final RequestPrimitiveDTO.Operation Notify
```

Method Detail

values

```
public static RequestPrimitiveDTO.Operation[] values()
```

valueOf

```
public static RequestPrimitiveDTO.Operation valueOf(String name)
```

getValue

```
public int getValue()
```

Enum RequestPrimitiveDTO.ResultContent

org.osgi.service.onem2m.dto

```
java.lang.Object
 └java.lang.Enum<RequestPrimitiveDTO.ResultContent>
   └org.osgi.service.onem2m.dto.RequestPrimitiveDTO.ResultContent
```

All Implemented Interfaces:

Comparable<RequestPrimitiveDTO.ResultContent>, Serializable

Enclosing class:

RequestPrimitiveDTO

```
public static enum RequestPrimitiveDTO.ResultContent
extends Enum<RequestPrimitiveDTO.ResultContent>
```

Enum Constant Summary	*Page*
attributes	68
attributesAndChildResourceReferences	69
attributesAndChildResources	69
childResourceReferences	69
childResources	69
hierarchicalAddress	69
hierarchicalAddressAndAttributes	69
nothing	68
originalResource	69

Method Summary		*Page*
int	**getValue**()	69
static RequestPrimitiveDTO.ResultContent	**valueOf**(String name)	69
static RequestPrimitiveDTO.ResultContent[]	**values**()	69

Enum Constant Detail

nothing

```
public static final RequestPrimitiveDTO.ResultContent nothing
```

attributes

```
public static final RequestPrimitiveDTO.ResultContent attributes
```

hierarchicalAddress

`public static final RequestPrimitiveDTO.ResultContent hierarchicalAddress`

hierarchicalAddressAndAttributes

`public static final RequestPrimitiveDTO.ResultContent hierarchicalAddressAndAttributes`

attributesAndChildResources

`public static final RequestPrimitiveDTO.ResultContent attributesAndChildResources`

attributesAndChildResourceReferences

`public static final RequestPrimitiveDTO.ResultContent attributesAndChildResourceReferences`

childResourceReferences

`public static final RequestPrimitiveDTO.ResultContent childResourceReferences`

originalResource

`public static final RequestPrimitiveDTO.ResultContent originalResource`

childResources

`public static final RequestPrimitiveDTO.ResultContent childResources`

Method Detail

values

`public static RequestPrimitiveDTO.ResultContent[] values()`

valueOf

`public static RequestPrimitiveDTO.ResultContent valueOf(String name)`

getValue

`public int getValue()`

Class ResourceDTO

org.osgi.service.onem2m.dto

```
java.lang.Object
 └org.osgi.dto.DTO
   └org.osgi.service.onem2m.dto.ResourceDTO
```

```
public class ResourceDTO
extends org.osgi.dto.DTO
```

DTO expressing Resource.

Field Summary		Page
Map<String,Object>	**attribute** Non Universal Attribute.	71
String	**creationTime**	71
List<String>	**labels**	71
String	**lastModifiedTime**	71
String	**parentID**	70
String	**resourceID**	70
String	**resourceName**	71
Integer	**resourceType**	70

Constructor Summary	Page
ResourceDTO()	71

Methods inherited from class org.osgi.dto.DTO
toString

Field Detail

resourceType

```
public Integer resourceType
```

resourceID

```
public String resourceID
```

parentID

```
public String parentID
```

creationTime

```
public String creationTime
```

lastModifiedTime

```
public String lastModifiedTime
```

resourceName

```
public String resourceName
```

labels

```
public List<String> labels
```

attribute

```
public Map<String,Object> attribute
```

Non Universal Attribute. Value Part must be the types that are allowed for OSGi DTO.

Constructor Detail

ResourceDTO

```
public ResourceDTO()
```

Enum ResourceType

org.osgi.service.onem2m.dto

```
java.lang.Object
 └java.lang.Enum<ResourceType>
   └org.osgi.service.onem2m.dto.ResourceType
```

All Implemented Interfaces:

Comparable<ResourceType>, Serializable

```
public enum ResourceType
extends Enum<ResourceType>
```

Enum Constant Summary	*Page*
`accessControlPolicy`	73
`accessControlPolicyAnnc`	77
`AE`	73
`AEAnnc`	77
`container`	73
`containerAnnc`	77
`contentInstance`	74
`contentInstanceAnnc`	77
`CSEBase`	74
`delivery`	74
`dynamicAuthorizationConsultation`	77
`dynamicAuthorizationConsultationAnnc`	78
`eventConfig`	74
`execInstance`	74
`fanOutPoint`	78
`flexContainer`	76
`flexContainerAnnc`	78
`group`	74
`groupAnnc`	77
`latest`	78
`locationPolicy`	74
`locationPolicyAnnc`	77
`m2mServiceSubscriptionProfile`	74
`mgmtCmd`	74
`mgmtObj`	74
`mgmtObjAnnc`	77
`node`	75
`nodeAnnc`	77
`notificationTargetMgmtPolicyRef`	76
`notificationTargetPolicy`	76
`oldest`	78
`policyDeletionRules`	76

pollingChannel	75
pollingChannelURI	79
remoteCSE	75
remoteCSEAnnc	77
request	75
role	76
schedule	75
scheduleAnnc	78
semanticDescriptor	76
semanticDescriptorAnnc	78
serviceSubscribedAppRule	75
serviceSubscribedNode	75
statsCollect	75
statsConfig	75
subscription	75
timeSeries	76
timeSeriesAnnc	78
timeSeriesInstance	76
timeSeriesInstanceAnnc	78
token	76
trafficPattern	76
trafficPatternAnnc	78

Method Summary		*Page*
int	**getValue**()	79
static ResourceType	**valueOf**(String name)	79
static ResourceType[]	**values**()	79

Enum Constant Detail

accessControlPolicy

`public static final ResourceType` **`accessControlPolicy`**

AE

`public static final ResourceType` **`AE`**

container

`public static final ResourceType` **`container`**

contentInstance

```
public static final ResourceType contentInstance
```

CSEBase

```
public static final ResourceType CSEBase
```

delivery

```
public static final ResourceType delivery
```

eventConfig

```
public static final ResourceType eventConfig
```

execInstance

```
public static final ResourceType execInstance
```

group

```
public static final ResourceType group
```

locationPolicy

```
public static final ResourceType locationPolicy
```

m2mServiceSubscriptionProfile

```
public static final ResourceType m2mServiceSubscriptionProfile
```

mgmtCmd

```
public static final ResourceType mgmtCmd
```

mgmtObj

```
public static final ResourceType mgmtObj
```

node

```
public static final ResourceType node
```

pollingChannel

```
public static final ResourceType pollingChannel
```

remoteCSE

```
public static final ResourceType remoteCSE
```

request

```
public static final ResourceType request
```

schedule

```
public static final ResourceType schedule
```

serviceSubscribedAppRule

```
public static final ResourceType serviceSubscribedAppRule
```

serviceSubscribedNode

```
public static final ResourceType serviceSubscribedNode
```

statsCollect

```
public static final ResourceType statsCollect
```

statsConfig

```
public static final ResourceType statsConfig
```

subscription

```
public static final ResourceType subscription
```

semanticDescriptor

```
public static final ResourceType semanticDescriptor
```

notificationTargetMgmtPolicyRef

```
public static final ResourceType notificationTargetMgmtPolicyRef
```

notificationTargetPolicy

```
public static final ResourceType notificationTargetPolicy
```

policyDeletionRules

```
public static final ResourceType policyDeletionRules
```

flexContainer

```
public static final ResourceType flexContainer
```

timeSeries

```
public static final ResourceType timeSeries
```

timeSeriesInstance

```
public static final ResourceType timeSeriesInstance
```

role

```
public static final ResourceType role
```

token

```
public static final ResourceType token
```

trafficPattern

```
public static final ResourceType trafficPattern
```

dynamicAuthorizationConsultation

```
public static final ResourceType dynamicAuthorizationConsultation
```

accessControlPolicyAnnc

```
public static final ResourceType accessControlPolicyAnnc
```

AEAnnc

```
public static final ResourceType AEAnnc
```

containerAnnc

```
public static final ResourceType containerAnnc
```

contentInstanceAnnc

```
public static final ResourceType contentInstanceAnnc
```

groupAnnc

```
public static final ResourceType groupAnnc
```

locationPolicyAnnc

```
public static final ResourceType locationPolicyAnnc
```

mgmtObjAnnc

```
public static final ResourceType mgmtObjAnnc
```

nodeAnnc

```
public static final ResourceType nodeAnnc
```

remoteCSEAnnc

```
public static final ResourceType remoteCSEAnnc
```

scheduleAnnc

```
public static final ResourceType scheduleAnnc
```

semanticDescriptorAnnc

```
public static final ResourceType semanticDescriptorAnnc
```

flexContainerAnnc

```
public static final ResourceType flexContainerAnnc
```

timeSeriesAnnc

```
public static final ResourceType timeSeriesAnnc
```

timeSeriesInstanceAnnc

```
public static final ResourceType timeSeriesInstanceAnnc
```

trafficPatternAnnc

```
public static final ResourceType trafficPatternAnnc
```

dynamicAuthorizationConsultationAnnc

```
public static final ResourceType dynamicAuthorizationConsultationAnnc
```

latest

```
public static final ResourceType latest
```

oldest

```
public static final ResourceType oldest
```

fanOutPoint

```
public static final ResourceType fanOutPoint
```

pollingChannelURI

```
public static final ResourceType pollingChannelURI
```

Method Detail

values

```
public static ResourceType[] values()
```

valueOf

```
public static ResourceType valueOf(String name)
```

getValue

```
public int getValue()
```

Class ResourceWrapperDTO

org.osgi.service.onem2m.dto

```
java.lang.Object
 └org.osgi.dto.DTO
   └org.osgi.service.onem2m.dto.ResourceWrapperDTO
```

```
public class ResourceWrapperDTO
extends org.osgi.dto.DTO
```

DTO expressing ResourceWrapper.

Field Summary		*Page*
String	uri	80

Constructor Summary	*Page*
ResourceWrapperDTO()	80

Methods inherited from class org.osgi.dto.DTO
toString

Field Detail

uri

`public String uri`

Constructor Detail

ResourceWrapperDTO

`public ResourceWrapperDTO()`

Class ResponsePrimitiveDTO

org.osgi.service.onem2m.dto

```
java.lang.Object
 └org.osgi.dto.DTO
   └org.osgi.service.onem2m.dto.ResponsePrimitiveDTO
```

```
public class ResponsePrimitiveDTO
extends org.osgi.dto.DTO
```

DTO expressing Response Primitive.

Nested Class Summary		Page
static enum	ResponsePrimitiveDTO.ContentStatus	84

Field Summary		Page
List<LocalTokenIdAssignmentDTO>	assignedTokenIdentifiers	83
Boolean	AuthSignatureReqInfo	83
PrimitiveContentDTO	content	82
Integer	contentOffset	82
ResponsePrimitiveDTO.ContentStatus	contentStatus	82
String	eventCategory	82
String	from	82
String	originatingTimestamp	82
ReleaseVersion	releaseVersionIndicator	83
String	requestIdentifier	82
Integer	responseStatusCode	82
String	resultExpirationTimestamp	82
String	to	82
List<DasInfoDTO>	tokenReqInfo	83
String	vendorInformation	83

Constructor Summary	Page
ResponsePrimitiveDTO()	83

Methods inherited from class org.osgi.dto.DTO
toString

Field Detail

responseStatusCode

```
public Integer responseStatusCode
```

---

requestIdentifier

```
public String requestIdentifier
```

---

content

```
public PrimitiveContentDTO content
```

---

to

```
public String to
```

---

from

```
public String from
```

---

originatingTimestamp

```
public String originatingTimestamp
```

---

resultExpirationTimestamp

```
public String resultExpirationTimestamp
```

---

eventCategory

```
public String eventCategory
```

---

contentStatus

```
public ResponsePrimitiveDTO.ContentStatus contentStatus
```

---

contentOffset

```
public Integer contentOffset
```

assignedTokenIdentifiers

```
public List<LocalTokenIdAssignmentDTO> assignedTokenIdentifiers
```

tokenReqInfo

```
public List<DasInfoDTO> tokenReqInfo
```

AuthSignatureReqInfo

```
public Boolean AuthSignatureReqInfo
```

releaseVersionIndicator

```
public ReleaseVersion releaseVersionIndicator
```

vendorInformation

```
public String vendorInformation
```

Constructor Detail

ResponsePrimitiveDTO

```
public ResponsePrimitiveDTO()
```

Enum ResponsePrimitiveDTO.ContentStatus

org.osgi.service.onem2m.dto

```
java.lang.Object
 └java.lang.Enum<ResponsePrimitiveDTO.ContentStatus>
   └org.osgi.service.onem2m.dto.ResponsePrimitiveDTO.ContentStatus
```

All Implemented Interfaces:

Comparable<ResponsePrimitiveDTO.ContentStatus>, Serializable

Enclosing class:

ResponsePrimitiveDTO

```
public static enum ResponsePrimitiveDTO.ContentStatus
extends Enum<ResponsePrimitiveDTO.ContentStatus>
```

Enum Constant Summary	*Page*
FULL_CONTENT	84
PARTIAL_CONTENT	84

Method Summary		*Page*
static ResponsePrimitiveDTO.ContentStatus	**valueOf**(String name)	85
static ResponsePrimitiveDTO.ContentStatus[]	**values**()	84

Enum Constant Detail

PARTIAL_CONTENT

```
public static final ResponsePrimitiveDTO.ContentStatus PARTIAL_CONTENT
```

FULL_CONTENT

```
public static final ResponsePrimitiveDTO.ContentStatus FULL_CONTENT
```

Method Detail

values

```
public static ResponsePrimitiveDTO.ContentStatus[] values()
```

---

valueOf

```
public static ResponsePrimitiveDTO.ContentStatus valueOf(String name)
```

Class ResponseTypeInfoDTO

org.osgi.service.onem2m.dto

```
java.lang.Object
 └org.osgi.dto.DTO
   └org.osgi.service.onem2m.dto.ResponseTypeInfoDTO
```

```
public class ResponseTypeInfoDTO
extends org.osgi.dto.DTO
```

Expressing ResponseTypeInfo

Nested Class Summary		Page
static enum	ResponseTypeInfoDTO.ResponseType	87

Field Summary		Page
List<String>	notificationURI	86
ResponseTypeInfoDTO.ResponseType	responseTypeValue	86

Constructor Summary	Page
ResponseTypeInfoDTO()	86

Methods inherited from class org.osgi.dto.DTO
toString

Field Detail

responseTypeValue

```
public ResponseTypeInfoDTO.ResponseType responseTypeValue
```

notificationURI

```
public List<String> notificationURI
```

Constructor Detail

ResponseTypeInfoDTO

```
public ResponseTypeInfoDTO()
```

Enum ResponseTypeInfoDTO.ResponseType

org.osgi.service.onem2m.dto

```
java.lang.Object
 └java.lang.Enum<ResponseTypeInfoDTO.ResponseType>
   └org.osgi.service.onem2m.dto.ResponseTypeInfoDTO.ResponseType
```

All Implemented Interfaces:

Comparable<ResponseTypeInfoDTO.ResponseType>, Serializable

Enclosing class:

ResponseTypeInfoDTO

```
public static enum ResponseTypeInfoDTO.ResponseType
extends Enum<ResponseTypeInfoDTO.ResponseType>
```

Enum Constant Summary	Page
blockingRequest	87
flexBlocking	88
nonBlockingRequestAsynch	87
nonBlockingRequestSynch	87

Method Summary		Page
int	**getValue**()	88
static ResponseTypeInfoDTO.ResponseType	**valueOf**(String name)	88
static ResponseTypeInfoDTO.ResponseType[]	**values**()	88

Enum Constant Detail

nonBlockingRequestSynch

`public static final ResponseTypeInfoDTO.ResponseType nonBlockingRequestSynch`

nonBlockingRequestAsynch

`public static final ResponseTypeInfoDTO.ResponseType nonBlockingRequestAsynch`

blockingRequest

`public static final ResponseTypeInfoDTO.ResponseType blockingRequest`

flexBlocking

```
public static final ResponseTypeInfoDTO.ResponseType flexBlocking
```

Method Detail

values

```
public static ResponseTypeInfoDTO.ResponseType[] values()
```

valueOf

```
public static ResponseTypeInfoDTO.ResponseType valueOf(String name)
```

getValue

```
public int getValue()
```

Class SecurityInfoDTO

[org.osgi.service.onem2m.dto](#)

```
java.lang.Object
 └org.osgi.dto.DTO
   └org.osgi.service.onem2m.dto.SecurityInfoDTO
```

```
public class SecurityInfoDTO
extends org.osgi.dto.DTO
```

DTO expressing Security Info.

Field Summary		Page
Map<String,Object>	dasRequest	89
Map<String,Object>	dasResponse	89
byte[]	escertkeMessage	90
String	esprimObject	90
Map<String,Object>	esprimRandObject	90
Integer	securityInfoType	89

Constructor Summary	Page
SecurityInfoDTO()	90

Methods inherited from class org.osgi.dto.DTO
toString

Field Detail

securityInfoType

`public Integer securityInfoType`

dasRequest

`public Map<String,Object> dasRequest`

dasResponse

`public Map<String,Object> dasResponse`

esprimRandObject

```
public Map<String,Object> esprimRandObject
```

esprimObject

```
public String esprimObject
```

escertkeMessage

```
public byte[] escertkeMessage
```

Constructor Detail

SecurityInfoDTO

```
public SecurityInfoDTO()
```

Java API documentation generated with DocFlex/Doclet v1.6.1

DocFlex/Doclet is both a multi-format Javadoc doclet and a free edition of DocFlex/Javadoc. If you need to customize your Javadoc without writing a full-blown doclet from scratch, DocFlex/Javadoc may be the only tool able to help you! Find out more at www.docflex.com

8 Considered Alternatives

For posterity, record the design alternatives that were considered but rejected along with the reason for rejection. This is especially important for external/earlier solutions that were deemed not applicable.

8.1 Representation of DTO

8.1.1 JAXB generated Class

As alternative solution, utilization of generated Java classes by JAXB has been considered, since oneM2M provides well defined XSD for defining data format. With the following aspects, this approach is not applied.

Many classes: Currently 65 XSD files are defined in oneM2M specification and JAXB tool (xjc) generates more than 140 Java classes. Using many classes as interface could make specification more complicated than its nature.

No Uniqueness: Generated classes by xjc are not unique, because it is possible to customize generation processes.

Changeability: Depending on the version of oneM2M, XSD files differ. It is preferable to choose version independent API, as much as possible. oneM2M ensures any data can be converted to JSON and CBOR, so proposed approach can be used with out modification, even if XSD file would be changed.

8.1.2 Generic DTO

Genric DTO, which has Map<String, Obj> in the top, has been discussed in Gent meeting. But it seems bad usage of defining DTO.

8.1.3 Specific DTO

SpecificDTO definitions have been generated from XSD generated classes. The number of DTO exceeds 170 and Java doc pages are getting 300 pages. It is apparently too much to express data formats. So middle approach of generic DTO and specific DTO has been chosen.

8.2 Resource Types Expression

In DTO, enum was eagerly used for clear candidates of possible values. But resource types seems more fragile because new resource types could be easily added. So Integer was chosen for resource types.

8.3 Use of Annotation defined by JAXB in DTO

Currently annotations defined by JAXB was used in DTO. It was pointed out as confusing because it might give impression that it only support XML serialization. But it was kept in the definitions by following reasons.

1. Removing the annotations are easier than inserting.

2. It is informative to specify optionality.

New OSGI annotation specifying optionality could be possible, but it might take time because it should be published as Core specification and R7 just has released.

8.4 Service Property for Service Layer Interface

In previous version, property named as PoAforAE is supposed to attach to Service Layer Interface. The way application need to search dedicated service for the AE, however, is not easy for developers nor OSGi way. The property has been removed and matching will be done by using Service Factory and mapping of bundle location and appropriate PoA is known by the service implementation.

8.5 Service Property for Notification Listener Interface

In previous version, property named as PoA and AE-ID are supposed to attach to Notification Listener Interface.

To attache AE-ID, application need to check result of <AE> resource creation and know modified AE-ID by CSE, and attach it as a property. This is complicated procedure and it is not a good design choice requiring all applications to implement this manner. Regarding PoA, the design that AE does not need to care of PoA was chosen.

Implementation of Service Layer Interface should manage both information by monitoring method request for creating <AE> resource and modifying the response. [This part need to be confirmed by reference implementation.]

9 Security Considerations

Description of all known vulnerabilities this may either introduce or address as well as scenarios of how the weaknesses could be circumvented.

9.1 ProtocolBinding Service with secure protocol configuration

In case that ProtocolBinding Service uses secure protocols, it is expected to handle pre-shared key or certificate and other parameters. Those configuration could be very diverse. This is out of scope of this RFC and it is responsibility of bundle developer that provides ProtocolBindingService.

9.2 Binding of AE Core and Protocol Binding

Protocol Binding has identity information, such as a key or certificate, which represents AE, so that AE core MUST be bound to the right protocol service binding. It is implementation's responsibility of protocol binding, it SHOULD utilize Service Factory to determine calling entity.

10 Document Support

10.1 References

[1]. Bradner, S., Key words for use in RFCs to Indicate Requirement Levels, RFC2119, March 1997.

[2]. oneM2M TS-0001 Functional Architecture, http://onem2m.org/images/files/deliverables/Release2/TS-0001-%20Functional_Architecture-V2_10_0.pdf

[3]. oneM2M TS-0004 Service Layer Core Protocol, http://onem2m.org/images/files/deliverables/Release2/TS-0004_Service_Layer_Core_Protocol_V2_7_1.zip

[4]. oneM2M TS-0001 Functional Architecture Draft v3.11.0, http://www.onem2m.org/technical/published-drafts

[5]. oneM2M TS-0004 Service Layer Core Protocol Draft v3.7.0, http://www.onem2m.org/technical/published-drafts

[6]. XSD files for oneM2M, https://git.onem2m.org/PRO/XSD.git

[7]. Software Requirements & Specifications. Michael Jackson. ISBN 0-201-87712-0 (NOTE:Is this needed?)

Add references simply by adding new items. You can then cross-refer to them by chosing <Insert><Cross Reference><Numbered Item> and then selecting the paragraph. ***STATIC REFERENCES (I.E. BODGED) ARE NOT ACCEPTABLE, SOMEONE WILL HAVE TO UPDATE THEM LATER, SO DO IT PROPERLY NOW.***

10.2 Author's Address

Name	Hiroyuki Maeomichi
Company	NTT
Address	Midorimachi 3-9-11, Musashino, Tokyo, Japan
Voice	+81 422 59 4072
e-mail	maeomichi.hiroyuki@lab.ntt.co.jp

10.3 Acronyms and Abbreviations

CSE: Common Services Entity

AE: Application Entity

CBOR: Concise Binary Object Representation

10.4 End of Document