|  |
| --- |
|  |

|  |  |
| --- | --- |
| CHANGE REQUEST | |
| Meeting ID:\* | SDS 40 |
| Source:\* | Bob Flynn, Convida Wireless , Bob.Flynn@convidawireless.com |
| Date:\* | 2019-05-21 |
| Reason for Change/s:\* | Examine use of “NOT\_ACCEPTABLE” versus “BAD\_REQUEST” |
| CR against: Release\* | Rel-3 |
| CR against: WI\* | Active <Work Item number>  MNT maintenance / < Work Item number(optional)>  Is this a mirror CR? Yes  No  mirror CR number: (Note to Rapporteur - use latest agreed revision)  STE Small Technical Enhancements / < Work Item number (optional)>  Only ONE of the above shall be ticked |
| CR against: TS/TR\* | TS-0004V3.11.2 |
| Clauses \* |  |
| Type of change: \* | Editorial change  Bug Fix or Correction  Change to existing feature or functionality  New feature or functionality  Only ONE of the above shall be ticked |
| Other TS/TR(s) impacted | None |
| Post Freeze checking:\* | This CR contains only essential changes and corrections? YES  NO  This CR may break backwards compatibility with the last approved version of the TS? YES  NO |
| Template Version: January 2019 (do not modify) | |

**oneM2M Notice**

The document to which this cover statement is attached is submitted to oneM2M. Participation in, or attendance at, any activity of oneM2M, constitutes acceptance of and agreement to be bound by terms of the Working Procedures and the Partnership Agreement, including the Intellectual Property Rights (IPR) Principles Governing oneM2M Work found in Annex 1 of the Partnership Agreement.

GUIDELINES for Change Requests:

Provide an informative introduction containing the problem(s) being solved, and a summary list of proposals.

Each CR should contain changes related to only one particular issue/problem.

If this is a correction, and the change applies to previous releases, a separate “mirror CR” should be posted at the same time as this CR

Mirror CR: applies only when the text, including clause numbering are exactly the same.

Companion CR: applies when the change means the same but the baselines differ in some way (e.g. clause number).

Follow the principle of completeness, where all changes related to the issue or problem within a deliverable are simultaneously proposed to be made e.g. a change impacting 5 tables should not only include a proposal to change only 3 tables. Include any changes to references, definitions, and abbreviations in the same deliverable.

Follow the drafting rules.

All pictures must be editable.

Check spelling and grammar.

Use change bars for modifications.

The change should include the current and surrounding clauses to clearly show where a change is located and to provide technical context of the proposed change. Additions of complete clauses need not show surrounding clauses as long as the proposed clause number clearly shows where the proposed new clause is located.

Multiple changes in a single CR shall be clearly separated by horizontal lines with embedded text such as, start of change 1, end of change 1, start of new clause, end of new clause.

When subsequent changes are made to the content of a CR, then the accepted version should not show changes over changes. The accepted version of the CR should only show changes relative to the baseline approved text.

## Introduction

In TS-0004 the two errors are defined, “NOT\_ACCEPTABLE”(5207) and “BAD REQUEST” (4000)

Action A.39.2-1: Review all places (in TS-0004) where “NOT\_ACCEPTABLE” appears and decide if a different response code should be used (e.g. BAD\_REQUEST).

“BAD REQUEST” is in the originator error class while “NOT\_ACCEPTABLE” is in the receiver error class.

So, if the error is because the originator did something incorrect, "BAD\_REQUEST" should be used instead of “NOT\_ACCEPTABLE”.

“NOT\_ACCEPTABLE” is appropriate if the request would be acceptable under different circumstances of the resources/CSE.

I used the following colors to indicate a change is recommended, a change might be needed, a change is not recommended.

I have included changes for all recommended and might be needed occurrences of “NOT\_ACCEPTABLE” in the CR.

I list all of the occurrences of “NOT\_ACCEPTABLE” in TS-0004.

#### 7.3.2.1 Check the validity of received request primitive

* 1. If the receiver does not support the content format (i.e. type of serialization) requested by the originator, the request may be rejected with a ***Response Status Code*** indicating "NOT\_ACCEPTABLE" error.

#### 7.3.3.2 Check existence of the addressed resource

* 1. If the Hosting CSE does not support the content format (i.e. type of serialization) requested by the originator, the request shall be rejected with a ***Response Status Code*** indicating "NOT\_ACCEPTABLE" error.

#### 7.3.3.18.0 Semantic resource discovery (CHANGE 1)

* 1. After Recv-1.0 "Check the validity of received request primitive": check that the syntax of the ***semanticsFilter*** corresponds to a valid SPARQL query request [33]. If the ***semanticsFilter*** content does not correspond to a valid SPARQL query request, the Receiver shall generate a Response Status Code indicating a "NOT\_ACCEPTABLE" error.

##### 7.3.3.19.1 Approach-1: Semantic query with implicit scope (CHANGE 2)

* 1. After Recv-1.0 "Check the validity of received request primitive": check that the syntax of the ***semanticsFilter*** corresponds to a valid SPARQL query request [33]. If the ***semanticsFilter*** content does not correspond to a valid SPARQL query request, the Receiver shall generate a Response Status Code indicating a "NOT\_ACCEPTABLE" error.

##### 7.4.7.2.1 Create (contentInstance) (CHANGE 3)

***Receiver:***

Primitive specific operation on Recv-6.5 "Create/Update/Retrieve/Delete/Notify operation is performed" with the following additional operations.

1. The Hosting CSE shall check whether the size in bytes of the *content* attribute of the <contentInstance> resource is greater than *maxByteSize* of the targeted parent <container> resource.

a) If true, the Hosting CSE shall return the response primitive with a ***Response Status Code*** indicating "NOT\_ACCEPTABLE" error. Skip steps 2 and 3 below.

##### 7.4.34.2.1 Create <semanticDescriptor> (CHANGE 4)

Primitive specific operation on Recv-6.4 "Check validity of resource representation for the given resource type":

a) The Hosting CSE shall check that the *descriptor* attribute conforms to the syntax defined by the *descriptorRepresentation* attribute.

b) If the *descriptor* attribute does not conform, the Hosting CSE shall reject the request with a ***Response Status Code*** indicating a "NOT\_ACCEPTABLE" error .

c) The Hosting CSE shall reject the request with a ***Response Status Code*** indicating a "BAD\_REQUEST" error if the *descriptorRepresentation* attribute is set to "IRI".

##### 7.4.34.2.3 Update (CHANGE 5)

Primitive specific operation on Recv-6.4 "Check validity of resource representation for the given resource type":

a) If both *semanticOpExec* and *descriptor* attributes exist, the Receiver shall generate a ***Response Status Code*** indicating a "NOT\_ACCEPTABLE" error.

b) If *semanticOpExec* attribute exists in the Request check that the syntax of its content corresponds to a valid SPARQL query request [33]. If the content does not correspond to a valid SPARQL query request, the Receiver shall generate a ***Response Status Code*** indicating a "NOT\_ACCEPTABLE" error.

c) If the *descriptor* attribute exists in the Request, check that the syntax of its content conforms to the syntax defined by the *descriptorRepresentation* attribute. If the content does not conform, the Receiver shall reject the request with a ***Response Status Code*** indicating a "NOT\_ACCEPTABLE" error.

d) The Hosting CSE shall reject the request with a ***Response Status Code*** indicating a "BAD\_REQUEST" error if the *descriptorRepresentation* attribute is set to "IRI".

##### 7.4.35.2.2 Retrieve <semanticFanOutPoint (CHANGE 6)

Check that the syntax of the ***semanticsFilter*** corresponds to a valid SPARQL query request [33]. If the ***semanticsFilter*** does not correspond to a valid SPARQL query request, the Receiver shall generate a ***Response Status Code*** indicating a "NOT\_ACCEPTABLE" error.

##### 7.4.47.2.1 Create <ontology> (CHANGE 7)

b) If the *ontologyContent* attribute does not conform, the Hosting CSE shall reject the request with a ***Response Status Code*** indicating a "NOT\_ACCEPTABLE" error.

##### 7.4.47.2.3 Update <ontology> (CHANGE 8)

If both *semanticOpExec* and *ontologyContent* attributes exist, the Receiver shall generate a ***Response Status Code*** indicating a "NOT\_ACCEPTABLE" error.

b) If the *semanticOpExec* attribute exists in the Request check that the syntax of its content corresponds to a valid SPARQL query request [33]. If the content does not correspond to a valid SPARQL query request, the Receiver shall reject the Request with a ***Response Status Code*** indicating a "NOT\_ACCEPTABLE" error.

c) If the *ontologyContent* attribute exists in the Request, check that the syntax of its content conforms to the syntax specified by the *ontologyFormat* attribute. If the content does not conform, the Receiver shall reject the Request with a ***Response Status Code*** indicating a "NOT\_ACCEPTABLE" error.

##### 7.4.49.2.1 Create <semanticMashupJobProfile (CHANGE 9)

If any of those attributes does not conform, the Hosting CSE shall generate a Response Status Code indicating a "NOT\_ACCEPTABLE" error.

##### 7.4.49.2.3 Update <semanticMashupJobProfile (CHANGE 10)

If any of those attributes does not conform, the Hosting CSE shall generate a Response Status Code indicating a "NOT\_ACCEPTABLE" error.

##### 7.4.50.2.1 Create <semanticMashupInstance(CHANGE 11)

c) If any of the above checks fail, the Hosting CSE shall generate a ***Response Status Code*** indicating a "NOT\_ACCEPTABLE" error.

##### 7.5.1.2.19 Notification for Subscription Blocking Triggered update (CHANGE 12)

If the notification ***Response Status Code*** is not successful, return a response to the original blocked UPDATE request primitive with a ***Response Status Code*** indicating NOT\_ACCEPTABLE.

### -----------------------Start of change 1-------------------------------------------

##### 7.3.3.18.0 Introduction

Semantic resource discovery is used to find resources in a CSE based on the semantic descriptions contained in the *descriptor* attribute of <*semanticDescriptor*> resources. Since an overall semantic description (forming a graph [i.5]) may be distributed across a set of <semanticDescriptor> resources, the semantic descriptions have to be retrieved (before or as needed) during the execution of the discovery request.

Semantic resource discovery is initiated by sending a Retrieve request with the discovery criteria in the ***semanticsFilter***filter condition(s) with two alternatives:

1. Targeting a <*semanticFanOutPoint>* virtual resource, see clause 7.4.35.
2. Targeting a resource other than <semanticFanOutPoint>. In this alternative the semantic resource discovery request procedure shall be comprised of the following actions:

***Originator:***

The Originator shall follow the steps from Orig-1.0 to Orig-6.0 specified in clause 7.2.2.1 Generic Resource Request Procedure for Originator.

In addition to Orig-1.0, the following steps shall be performed.

The *To* parameter in the Retrieve Request shall indicate the root of where the semantic discovery begins.

The *filterCriteria* of the Retrieve Request shall include the *filterUsage* parameter configured as "discovery" and the ***semanticsFilter*** filter condition.

***Receiver:***

The Receiver shall follow the steps from Recv-1.0 to Recv-7.0 specified in clause 7.2.2.2 Generic Resource Request Procedure for Receiver.

After Recv-1.0 "Check the validity of received request primitive": check that the syntax of the ***semanticsFilter*** corresponds to a valid SPARQL query request [33]. If the ***semanticsFilter*** content does not correspond to a valid SPARQL query request, the Receiver shall generate a Response Status Code indicating a "BAD\_REQUEST" error.

The Hosting CSE shall follow the steps from Recv-1.0 to Recv-6.2 specified in clause 7.2.2.2.The Hosting CSE shall not perform steps from Recv-6.3 to Recv-6.6 and perform the following steps instead:

1. The Hosting CSE shall find the <semanticDescriptor> resource(s) to which the Originator has "Discover" access right, under the addressed resource.

a) If the *relatedSemantics* attribute does not exist, the "Annotation-based method" (using *resourceDescriptorLink)* detailed in clause 7.3.3.18.1 shall be used.

b) If the *relatedSemantics* attribute exists the "Resource link-based method" (using the *relatedSemantics attribute*)detailed in clause 7.3.3.18.2 shall be used.

1. The Hosting CSE shall perform Recv-6.7 "Create a success response" where the Response shall include the resources matched based on the SPARQL engine result.

### -----------------------End of change 1---------------------------------------------

### -----------------------Start of change 2-------------------------------------------

##### 7.3.3.19.1 Approach-1: Semantic query with implicit scope

In Approach-1, a semantic query request message targets any resource (i.e. as specified by the ***To*** parameter) and the semantic query shall be executed relative to this target resource, similarly to other request messages. The scope of the semantic query is formed through the aggregation of the semantic contents of the target resource's descendants. All the contents of semantic resource descendants of the target resource shall form the RDF data basis for this semantic query to be executed on. In this alternative, the semantic query procedure shall be comprised of the following actions:

***Originator:***

The Originator shall follow the steps from Orig-1.0 to Orig-6.0 specified in clause 7.2.2.1 Generic Resource Request Procedure for Originator.

In addition to Orig-1.0, the following steps shall be performed.

The ***To*** parameter in the Retrieve Request shall define the scope of this semantic query as mentioned earlier.

The Retrieve Request shall include the following parameters:

1. the ***Semantic Query Indicator***, which is set to true;
2. ***filterCriteria*** of the Retrieve Request shall include the ***semanticsFilter*** condition tag; and
3. the parameter Result Content shall be set to "semantic content" to indicate that the response message shall contain the result of a semantic query.

***Receiver:***

The Receiver shall follow the steps from Recv-1.0 to Recv-7.0 specified in clause 7.2.2.2 Generic Resource Request Procedure for Receiver.

After Recv-1.0 "Check the validity of received request primitive": check that the syntax of the ***semanticsFilter*** corresponds to a valid SPARQL query request [33]. If the ***semanticsFilter*** content does not correspond to a valid SPARQL query request, the Receiver shall generate a Response Status Code indicating a "BAD\_REQUEST" error.

The Hosting CSE shall follow the steps from Recv-1.0 to Recv-6.2 specified in clause 7.2.2.2. The Hosting CSE shall not perform steps from Recv-6.3 to Recv-6.6 and perform the following steps instead:

1. The Hosting CSE shall find the semantic resources to which the Originator has "RETRIEVE" access right, under the addressed resource as specified by the ***To*** parameter.
2. Aggregate the semantic resources and deliver the content for SPARQL processing, along with the ***semanticsFilter*** content.
3. Wait for a SPARQL processing response.
4. Perform Recv-6.7 "Create a success response" where the Response shall include the SPARQL processing result, which is the semantic query result to be returned.
5. Perform Recv-6.8 and the procedure is terminated.

### -----------------------End of change 2---------------------------------------------

### -----------------------Start of change 3-------------------------------------------

##### 7.4.7.2.1 Create

***Originator:***

No change from the generic procedures in clause 7.2.2.1.

***Receiver:***

Primitive specific operation on Recv-6.5 "Create/Update/Retrieve/Delete/Notify operation is performed" with the following additional operations.

1. The Hosting CSE shall check whether the size in bytes of the *content* attribute of the <contentInstance> resource is greater than *maxByteSize* of the targeted parent <container> resource.

a) If true, the Hosting CSE shall return the response primitive with a ***Response Status Code*** indicating "NOT\_ACCEPTABLE" error. Skip steps 2 and 3 below.

b) If false, the Hosting CSE shall set the *contentSize* attribute of the <contentInstance> resource to the size in bytes of the *content* attribute.

1. The Hosting CSE shall check the *currentNrOfInstances* and *currentByteSize* of the targeted parent <container> resource.

a) If *maxNrOfInstances* of the targeted parent <container> resource is specified then if the *currentNrOfInstances* when modified to reflect the addition of the new *<*contentInstance*>* exceeds *maxNrOfInstances*, the Hosting CSE shall remove the oldest *<*contentInstance*>* resource from the targeted <container*>* resource.

b) If *maxByteSize* of the targeted parent <container> resource is specified then if the *currentByteSize* when modified to reflect the addition of the new *<*contentInstance*>* exceeds *maxByteSize* the Hosting CSE shall remove the oldest *<*contentInstance*>* resources from the targeted *<*container*>* resource until *maxByteSize* conditions are met.

c) The Hosting CSE shall update the *currentNrOfInstances* of the targeted parent <container> resource with the count of <contentInstance> resources in the targeted parent <container> resource. The Hosting CSE shall update the *currentByteSize* of the targeted parent <container> resource with the sum of the *contentSize* attributes of the <contentInstance> resources in the targeted parent <container> resource.

d) When removing the oldest *<*contentInstance*>* resources, the Hosting CSE shall not generate notifications even if there exists a <subscription> to the targeted <container> resource and this <subscription> is configured to generate a notification on "Delete\_of\_Direct\_Child\_Resource".

e) If the *maxInstanceAge* attribute is present in the targeted parent <container> resource, then the Hosting CSE shall set the *expirationTime* attribute in <contentInstance> resource such that the time difference between *expirationTime* and the *creationTime* of the <contentInstance> resource shall not exceed the *maxInstanceAge* of the targeted parent <container> resource.

1. The Hosting CSE shall increment the *stateTag* attribute of the targeted parent <container> resource and copy the value into the *stateTag* attribute of the <contentInstance> resource.
2. If the hosting CSE has the capability to duplicate the actual data in semantic triples, it may decide whether to represent the *content* as semantic triples, depending on local policies/configurations. If the hosting CSE decides to do so, it shall execute the following actions: a) represent the actual data contained in the *content* attribute to semantic triples (e.g. RDF triples); b) create a <semanticDescriptor> child resource for the <contentInstance> resource with its descriptor attribute set to these semantic triples generated in a).
3. If the hosting CSE does not have the capability to duplicate the actual data in semantic triples complying with an ontology that it supports, this step will be skipped.

No other changes from the generic procedures in clause 7.2.2.2.

### -----------------------End of change 3---------------------------------------------

### -----------------------Start of change 4-------------------------------------------

##### 7.4.34.2.1 Create

***Originator:***

No change from the generic procedures in clause 7.2.2.1.

***Receiver:***

No change from the generic procedures in clause 7.2.2.2 with the following exception:

1. Primitive specific operation on Recv-6.4 "Check validity of resource representation for the given resource type":

a) The Hosting CSE shall check that the *descriptor* attribute conforms to the syntax defined by the *descriptorRepresentation* attribute.

b) If the *descriptor* attribute does not conform, the Hosting CSE shall reject the request with a ***Response Status Code*** indicating a "BAD\_REQUEST" error .

c) The Hosting CSE shall reject the request with a ***Response Status Code*** indicating a "BAD\_REQUEST" error if the *descriptorRepresentation* attribute is set to "IRI".

1. Primitive specific operation on Recv-6.5 "Create/Update/Retrieve/Delete/Notify operation is performed":

a) The Hosting CSE shall set the *validationEnable* attribute of the <semanticDescriptor> resource based on the value provided in the request and its local policy. Note that the local policy may override the suggested value provided in the request from the originator to enforce or disable the following semantic validation procedures. There are different cases depending on how the local policy is configured (which is out of the scope of the present document) and whether/how the *validationEnable* attribute is provided in the request:

* *validationEnable* attribute is not present if it was not provided in the request or if the local policy does not allow for the *validationEnable* attribute;
* *validationEnable* attribute is set to true or false according to the local policy no matter how the value is provided in the request;
* *validationEnable* attribute is set to true or false according to the value provided in the request.

b) If the *validationEnable* attribute is set as true, the hosting CSE shall perform the semantic validation process in the following steps according to clause 7.10.2 in oneM2M TS-0034 [50]. Otherwise, skip the following steps.

c) Check if the addressed *<semanticDescriptor>* resource is linked to other *<semanticDescriptor>* resources on a remote CSE by the *relatedSemantics* attribute or by triples with annotation property *m2m:resourceDescriptorLink* in *descriptor* attribute. This process shall consider the recursive links.

* If yes, the Hosting CSE shall generate an Update request primitive with itself as the Originator and with the ***Content*** parameter set to the addressed *<semanticDescriptor>* resource representation, and send it to the <semanticValidation> virtual resource URI on the CSE which hosts the referenced ontology (following the *ontologyRef* attribute) of the addressed *<semanticDescriptor>* resource (see details in clause 7.4.48.2.3). After receiving the response primitive, i.e. the validation result, go to step k. If no response primitive was received due to time-out or other exceptional cases, the hosting CSE shall generate a ***Response Status Code*** indicating a "TARGET\_NOT\_REACHABLE" error.
* If no, perform the following steps.

d) Access the semantic triples from the *descriptor* attribute of the received <semanticDescriptor> resource.

e) Access the ontology referenced in the *ontologyRef* attribute of the received <semanticDescriptor> resource.

* If the ontology referenced by the *ontologyRef* attribute is an external ontology, not locally hosted by the Hosting CSE, the Hosting CSE shall retrieve it using the corresponding protocol and identifier information specified in the *ontologyRef* attribute.
* If the referenced ontology cannot be retrieved within a reasonable time (as defined by a local policy), the Hosting CSE shall generate a ***Response Status Code*** indicating an "ONTOLOGY\_NOT\_AVAILABLE" error.

f) Retrieve any local linked <semanticDescriptor> resources of the received <semanticDescriptor> resource following the URI(s) in the *relatedSemantics* attribute (if it exists) and the URI(s) in the triples with annotation property m2m:resourceDescriptorLink (if there are any).

* Repeat this step recursively to Retrieve any further local linked <semanticDescriptor> resources.
* If the local linked <semanticDescriptor> resources cannot be retrieved within a reasonable time (which is subject to a local policy), the Hosting CSE shall generate a ***Response Status Code*** indicating a "LINKED\_SEMANTICS\_NOT\_AVAILABLE" error.

g) Retrieve the semantic triples from the *descriptor* attribute of the local linked <semanticDescriptor> resource.

h) Retrieve the referenced ontologies of the local linked <semanticDescriptor> resources following the URI(s) in *ontologyRef* attribute of the linked <semanticDescriptor> resources; If the referenced ontologies cannot be retrieved within a reasonable time (as defined by a local policy), the Hosting CSE shall generate a ***Response Status Code*** indicating an "ONTOLOGY\_NOT\_AVAILABLE" error.

i) Combine all the semantic triples of the addressed and local linked <semanticDescriptor> resources as the set of semantic triples to be validated, and combine all the referenced ontologies as the set of ontologies to validate the semantic triples against.

j) Check all the aspects of semantic validation according to clause 7.10.3 in oneM2M TS-0034 [50] based upon the semantic triples and referenced ontology. If any problem occurs, the Hosting CSE shall generate a ***Response Status Code*** indicating an "INVALID\_SEMANTICS" error.

k) After the semantic validation process, the Hosting CSE shall set the *semanticValidated* attribute of the addressed <semanticDescriptor> resource according to the validation result (i.e. set to true if the no error occurs until now, otherwise false).

l) Based on its local policy, the Hosting CSE may also update the value of the *semanticValidated* attributes of the local linked <semanticDescriptor> resources according to the validation result.

### -----------------------End of change 4---------------------------------------------

### -----------------------Start of change 5-------------------------------------------

##### 7.4.34.2.3 Update

***Originator:***

No change from the generic procedures in clause 7.2.2.1 with the following exception:

* The *descriptor* attribute can be updated using SPARQL as follows:

Primitive specific operation on Orig-1.0 "Compose Request primitive": The originator creates a request to update the *semanticOpExec* attribute. The value of this attribute is set to a SPARQL request that includes INSERT, DELETE, or DELETE/INSERT with conditional SPARQL statements as defined in the SPARQL query language [33].

***Receiver:***

No change from the generic procedures in clause 7.2.2.2 with the following exceptions:

1. Primitive specific operation on Recv-6.4 "Check validity of resource representation for the given resource type":

a) If both *semanticOpExec* and *descriptor* attributes exist, the Receiver shall generate a ***Response Status Code*** indicating a "BAD\_REQUEST" error.

b) If *semanticOpExec* attribute exists in the Request check that the syntax of its content corresponds to a valid SPARQL query request [33]. If the content does not correspond to a valid SPARQL query request, the Receiver shall generate a ***Response Status Code*** indicating a "BAD\_REQUEST" error.

c) If the *descriptor* attribute exists in the Request, check that the syntax of its content conforms to the syntax defined by the *descriptorRepresentation* attribute. If the content does not conform, the Receiver shall reject the request with a ***Response Status Code*** indicating a "BAD\_REQUEST" error.

d) The Hosting CSE shall reject the request with a ***Response Status Code*** indicating a "BAD\_REQUEST" error if the *descriptorRepresentation* attribute is set to "IRI".

1. Primitive specific operation on Recv-6.5 "Create/Update/Retrieve/Delete/Notify operation is performed" in addition:

a) If *semanticOpExec* attribute exists in the Request, the Hosting CSE shall update the semantic triples in the *descriptor* attribute according to SPARQL update request in the *semanticOpExec* attribute. If the SPARQL update request cannot be executed, the Hosting CSE shall "create an unsuccessful Response primitive" with the ***Response Status Code*** indicating "SPARQL\_UPDATE\_ERROR", otherwise proceed to step Recv-6.6.

b) The hosting CSE shall set the *validationEnable* attribute of the addressed <semanticDescriptor> resource based on the value provided in the request and its local policy. Note that the local policy may override the suggested value provided in the request from the originator to enforce or disable the following semantic validation procedures. There are different cases depending on how the local policy is configured (which is out of the scope of the present document) and whether/how the *validationEnable* attribute is provided in the request:

* no change to the existing *validationEnable* attribute if it is not provided in the request;
* *validationEnable* attribute is not present if the local policy does not allow for the *validationEnable* attribute;
* *validationEnable* attribute is set to true or false according to the local policy no matter how the value is provided in the request;
* *validationEnable* attribute is set to true or false according to the value provided in the request.

c) The hosting CSE shall perform steps 2b-2l as specified in clause 7.4.34.2.1.

d) If *validationEnable* attribute is changed from true to false, then the hosting CSE shall set the *semanticValidated* attribute of the addressed <semanticDescriptor> resource as false.

### -----------------------End of change 5---------------------------------------------

### -----------------------Start of change 6-------------------------------------------

##### 7.4.35.2.2 Retrieve

***Originator:***

No primitive specific operations.

***Receiver:***

The Receiver shall follow the steps from Recv-1.0 to Recv-6.2 specified in clause 7.2.2.2 Generic Resource Request Procedure for Receiver, with the following primitive specific operations:

After Recv-1.0 "Check the validity of received request primitive":

1. Check that the syntax of the ***semanticsFilter*** corresponds to a valid SPARQL query request [33]. If the ***semanticsFilter*** does not correspond to a valid SPARQL query request, the Receiver shall generate a ***Response Status Code*** indicating a "BAD\_REQUEST" error.
2. If the ***Semantic Query Indicator*** parameter included in the request message is set to true, the request shall be processed as a semantic query. Otherwise, the request shall be processed as a semantic resource discovery.

After Recv-6.2 "Check existence of the addressed resource":

1. Check that the *semanticSupportIndicator* of the parent <group> resource is set to true.
2. Check the authorization of the Originator using the *membersAccessControlPolicyIDs* of the parent group resource. In the case the *membersAccessControlPolicyIDs* is not provided, the *accessControlPolicyIDs* of the parent group resource shall be used.
3. Fan-out <semanticDescriptor> Retrieve Requests to each CSE hosting sub-groups or members as follows:

For each group member, the Hosting CSE shall perform the following steps:

1. The primitive parameters ***From*** and ***To*** shall be mapped to corresponding Retrieve Requests to be sent out to each member of the group. The primitive parameter From shall be used directly. The prefix of primitive parameter ***To*** i.e. <URI of group resource>/sfop shall be replaced by hierarchical URIs derived from the attribute *memberIDs* of the <group> resource.
2. The group hosting CSE shall execute "Compose Request primitives" with the ***semanticsFilter*** filter condition set to false.
3. "Send the Request to the receiver CSE".
4. "Wait for Response primitive".
5. Once the Responses to the Retrieve Requests have been received, proceed to the following steps:
6. Aggregate the descriptors from the Retrieve Responses and deliver the content for SPARQL processing, along with the ***semanticsFilter*** content.
7. Wait for a SPARQL processing response.
8. Perform Recv-6.7 "Create a success response" where the Response shall include the SPARQL processing result. In case of semantic query, the Response shall include the semantic query result. In case of semantic resource discovery, the Response shall include a list of identified resource URIs.
9. Perform Recv-6.8 and the procedure is terminated.

### -----------------------End of change 6---------------------------------------------

### -----------------------Start of change 7-------------------------------------------

##### 7.4.47.2.1 Create

***Originator:***

No change from the generic procedures in clause 7.2.2.1.

***Receiver:***

No change from the generic procedures in clause 7.2.2.2 with the following exception:

1. Primitive specific operation on Recv-6.4 "Check validity of resource representation for the given resource type":

a) The Hosting CSE shall check that the *ontologyContent* attribute conforms to the syntax defined by the *ontologyFormat* attribute.

b) If the *ontologyContent* attribute does not conform, the Hosting CSE shall reject the request with a ***Response Status Code*** indicating a "BAD\_REQUEST" error.

### -----------------------End of change 7---------------------------------------------

### -----------------------Start of change 8-------------------------------------------

##### 7.4.47.2.3 Update

***Originator:***

No change from the generic procedures in clause 7.2.2.1 with the following exception:

1. Primitive specific operation on Orig-1.0 "Compose Request primitive": The originator creates a request to update the *semanticOpExec* attribute. The value of this attribute is set to a SPARQL request that includes INSERT, DELETE, or DELETE/INSERT with conditional SPARQL statements as defined in the SPARQL query language [33].

***Receiver:***

No change from the generic procedures in clause 7.2.2.2 with the following exceptions:

1. Primitive specific operation on Recv-6.4 "Check validity of resource representation for the given resource type":

a) If both *semanticOpExec* and *ontologyContent* attributes exist, the Receiver shall generate a ***Response Status Code*** indicating a "BAD\_REQUEST" error.

b) If the *semanticOpExec* attribute exists in the Request check that the syntax of its content corresponds to a valid SPARQL query request [33]. If the content does not correspond to a valid SPARQL query request, the Receiver shall reject the Request with a ***Response Status Code*** indicating a "BAD\_REQUEST" error.

c) If the *ontologyContent* attribute exists in the Request, check that the syntax of its content conforms to the syntax specified by the *ontologyFormat* attribute. If the content does not conform, the Receiver shall reject the Request with a ***Response Status Code*** indicating a "BAD\_REQUEST" error.

1. Primitive specific operation on Recv-6.5 "Create/Update/Retrieve/Delete/Notify operation is performed" in addition:

a) If the *semanticOpExec* attribute exists in the Request, the Hosting CSE shall update the semantic triples in the *ontologyContent* attribute according to the SPARQL update request in the s*emanticOpExec* attribute. If the SPARQL update request cannot be executed, the Hosting CSE shall "create an unsuccessful Response primitive" with the ***Response Status Code*** indicating "SPARQL\_UPDATE\_ERROR", otherwise proceed to step Recv-6.6.

### -----------------------End of change 8---------------------------------------------

### -----------------------Start of change 9-------------------------------------------

##### 7.4.49.2.1 Create

***Originator:***

No change from the generic procedures in clause 7.2.2.1.

***Receiver:***

No change from the generic procedures in clause 7.2.2.2 with the following exception:

1. Primitive specific operation on Recv-6.4 "Check validity of resource representation for the given resource type":

a) The Hosting CSE shall check that the *inputDescriptor, outputDescriptor* and *functionDescriptor* attributes conform to the RDF/XML syntax as defined in RDF 1.1 XML Syntax [34].

b) The hosting CSE shall also check that the *memberFilter* attribute conforms to a valid SPARQL query request [33].

c) If any of those attributes does not conform, the Hosting CSE shall generate a Response Status Code indicating a "BAD\_REQUEST" error.

### -----------------------End of change 9---------------------------------------------

### -----------------------Start of change 10-------------------------------------------

##### 7.4.49.2.3 Update

***Originator:***

No change from the generic procedures in clause 7.2.2.1.

***Receiver:***

No change from the generic procedures in clause 7.2.2.2 with the following exceptions:

1. Primitive specific operation on Recv-6.4 "Check validity of resource representation for the given resource type":

a) If any of those attributes (*inputDescriptor, outputDescriptor* and *functionDescriptor*) is being updated, The Hosting CSE shall check that the new values of those attributes being updated conform to the RDF/XML syntax as defined in RDF 1.1 XML Syntax [34].

b) If the *memberFilter* attribute is being updated, the hosting CSE shall check that the new value of the *memberFilter* attribute conforms to a valid SPARQL query request [33].

c) If any of the new values of those attributes does not conform, the Hosting CSE shall generate a ***Response Status Code*** indicating a "BAD\_REQUEST" error.

### -----------------------End of change 10---------------------------------------------

### -----------------------Start of change 11-------------------------------------------

##### 7.4.50.2.1 Create

***Originator:***

No change from the generic procedures in clause 7.2.2.1.

***Receiver:***

No change from the generic procedures in clause 7.2.2.2 with the following exceptions:

1. Primitive specific operation on Recv-6.4 "Check validity of resource representation for the given resource type":

a) The Hosting CSE shall check that the value of the *smjpID* attribute is not NULL and a valid SMJP can be retrieved by using the SMJP ID indicated in the *smjpID* attribute.

b) The Hosting CSE shall check that the value of *smjpInputParameter* attribute conforms to the RDF/XML syntax as defined in RDF 1.1 XML Syntax [34] and that it meets the requirement described in the "*inputDescriptor*" attribute of the retrieved <*semanticMashupJobProfile*>.

c) If any of the above checks fail, the Hosting CSE shall generate a ***Response Status Code*** indicating a "BAD\_REQUEST" error.

d) The Hosting CSE shall use the SPARQL query specified in the *memberFilter* attribute of the retrieved <*semanticMashupJobProfile*> to discover mashup member resources for the <*semanticMashupInstance*> to be created. The hosting CSE shall conduct semantic resource discovery on one or more CSEs by using the SPARQL query statement as the semantics filter to identify qualified mashup members.

e) If not all of the required mashup members can be identified, the Hosting CSE shall generate a ***Response Status Code*** indicating a "MASHUP\_MEMBER\_NOT\_FOUND" error.

f) If all the mashup members are identified, the *mashupMember* attribute shall be updated. If the *memberStoreType* attribute has the value of "URI\_ONLY", then the URIs of those member resources identified (during step d) will be stored in the *mashupMember* attribute. If the *memberStoreType* attribute has the value of "URI\_AND\_VALUE", then the URIs as well as the values of those member resources identified (during step d) will be stored in the *mashupMember* attribute.

1. Primitive specific operation on Recv-6.5 “Create/Update/Retrieve/Delete/Notify operation is performed”:

a) If the *resultGenType* is set to WHEN\_SMI\_IS\_CREATED the procedure for calculation of the semantic mashup result is performed and a new *<semanticMashupResult>* child resource is created.

b) If the *resultGenType* is set to PERIODICALLY, a timer is started with the given period, at which time the procedure for calculation of the semantic mashup result is performed and a new *<semanticMashupResult>* child resource is created.

### -----------------------End of change 11---------------------------------------------

### -----------------------Start of change 12-------------------------------------------

##### 7.5.1.2.19 Notification for Subscription Blocking Triggered update

Whenever the Hosting CSE receives an update request primitive for a target resource which has subscription with *notificationEventType* set to "Blocking\_Update", it shall perform the steps listed below before Recv-6.5 "Create/Update/Retrieve/Delete/Notify operation” is performed.

1. If the *attribute* condition tag of the *eventNotificationCriteria* attribute of the <subscription> resource is set, check that the *attribute* condition tag matches the modified attributes in the received UPDATE request.
2. Prevent or block all other UPDATE request primitives to this target resource.
3. Create a Notification request primitive and configure the request parameters as follows.

a) Set the *representation* attribute of the notification to the representation of the target resource contained in the received UPDATE request primitive.

1. Send the Notification request primitive to the target specified in *notificationURI.*
2. Wait for a Notification response.
3. Process the Notification response primitive

a) If the notification ***Response Status Code*** is not successful, return a response to the original blocked UPDATE request primitive with a ***Response Status Code*** indicating NOT\_ACCEPTABLE.

b) If the notification ***Response Status Code*** is successful, perform Recv-6.5 "Create/Update/Retrieve/Delete/Notify operation” for the received UPDATE request.

1. Allow all other UPDATE request primitives for this target resource.

### -----------------------End of change 12---------------------------------------------

### -----------------------Start of change 2-------------------------------------------

### -----------------------End of change 2---------------------------------------------

### -----------------------Start of change 2-------------------------------------------

### -----------------------End of change 2---------------------------------------------

CHECK LIST

* Does this Change Request include an informative introduction containing the problem(s) being solved, and a summary list of proposals.?
* Does this CR contain changes related to only one particular issue/problem?
* Have any mirror CRs been posted?
* Does this Change Request make **all** the changes necessary to address the issue or problem? E.g. A change impacting 5 tables should not include a proposal to change only 3 tables?Does this Change Request follow the drafting rules?
* Are all pictures editable?
* Have you checked the spelling and grammar?
* Have you used change bars for all modifications?
* Does the change include the current and surrounding clauses to clearly show where a change is located and to provide technical context of the proposed change? (Additions of complete clauses need not show surrounding clauses as long as the proposed clause number clearly shows where the new clause is proposed to be located.)
* Are multiple changes in this CR clearly separated by horizontal lines with embedded text such as, start of change 1, end of change 1, start of new clause, end of new clause.?