|  |  |
| --- | --- |
| CHANGE REQUEST | |
| Meeting ID:\* |  |
| Source:\* |  |
| Date:\* |  |
| Reason for Change/s:\* |  |
| CR against: Release\* |  |
| CR against: WI\* | Active <WI-0100>  MNT maintenance / < Work Item number(optional)>  Is this a mirror CR? Yes  No  mirror CR number:  STE Small Technical Enhancements / < Work Item number (optional)>  Only ONE of the above shall be ticked |
| CR against: TS/TR\* |  |
| 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.

In case of a correction, and the change apply to previous releases, a separate “mirror CR” should be posted at the same time of 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. Includes any changes to references, definitions, and acronyms in the same deliverable.

Follow the drafting rules.

All pictures must be editable.

Check spelling and grammar to the extent practicable.

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 new clause is proposed to be 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 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

wordings & revision

<https://git.onem2m.org/specifications/ts/ts-0041/-/merge_requests/11>

<https://git.onem2m.org/specifications/ts/ts-0041/-/merge_requests/11/diffs?commit_id=dc5f5c75bcdf4a6e55e7f4916ce68c9f32259a00>

========= Comment provided by Ingo Friese at 2025-04-03T05:59:11.543Z: =========

1. delete double word "the"

2. Since the notificationURI in the Subscription is an AE resourceID no verification takes place.

So take out all verification aspects

========= Comment provided by Peter Niblett at 2025-04-01T03:23:05.544Z: =========

Could you make the following changes please?

1. In 6.3.2.0 you currently have

> Furthermore the \*pointOfAccess\* (poa) attribute of the `<AE>` resource may be set to the address of the IPE where it can receive notifications from the CSE. Alternatively the \*notificationURI\* attribute may be set in the `<subscription>` resource creation request described in clause 6.3.2.1 .

These are not alternatives. You have to set the \*notificationURI\* attribute in the `<subscription>` resource. The only choice is whether you want to use oneM2M protocols to send the notification to the IPE (in which case you set the \*notificationURI\* to the `<AE>` resourceID), or whether you use some other mechanism, in which case you set the \*notificationURI\* to the URI used by that mechanism. I think that you should require the former, which means that you need to say that:

- The `<AE>` shall have \*requestReachability\* set to `true`

- The `<AE>` shall have a \*pointOfAccess\* attribute giving the protocol and address that the IPE is going to use to receive notifications from the CSE.

2. In 6.3.2.1 you currently have

> A `<subscription>` resource shall be created under this `<container>` resource. During its creation, a verification notification request is sent to the IPE. The IPE must respond with an HTTP status code of 200 and a oneM2M response status code of 2000.

If no \*pointOfAccess\* (poa) attribute is set in the `<AE>` resource, the \*notificationURI\* attribute in the `<subscription>` resource creation request shall be set to the address of the IPE.

- You should say here that when the subscription is created its \*notificationURI\* shall be set to the IPE's `<AE>` resourceID.

- The second sentence should say 'a verification notification \*\*may\*\* be sent to the IPE' rather than \*\*shall\*\* be sent since the hosting CSE is not required to send one.

- Consequently the third sentence should start 'If it receives a verification request, the IPE shall ...

- The third sentence must not mention HTTP since the protocol used is specified by the \*pointOfAccess\* in the `<AE>` and might not be HTTP. I would say 'If it receives a verification request, the IPE shall check the validity of this request and then reply with a oneM2M response containing a \*\*\_Response Status Code\_\*\* indicating "OK".'

- Delete the fourth sentence, since the \*pointOfAccess\* is always required (this is now covered in 6.3.2.0)

3. Please change validation to verification in this sentence:

> 2) The hosting CSE evaluates the request and performs the appropriate checks, including validation that the notification endpoint of the IPE is available. It then creates the `<subscription>` resource.

4. You have a sentence about handling the actual notifications when they come in

> Since the IPE has subscribed to this `<container>` resource it receives a notification message along with all attributes of the `<contentInstance>` resource when new data arrives.

Do you say anywhere what it does with this notification message when it arrives? It needs to check the validity of the notification, e.g. to check that it does come from the subscription that it created and isn't a spam message. Or are you planning to handle Security somewhere else in the TS.

5. There are a few places where you use the phrase "result message". In oneM2M we call this a Response or Response primitive.

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

---a/TS-0041-oneM2M-SensorThings\_interworking.md  
+++b/TS-0041-oneM2M-SensorThings\_interworking.md

@@ -244,6 +244,73 @@ Once the destination-"Datastream" is created, the IPE can send an "Observation"

\*\*Figure 6.3.1.2-1: Message flow from IPE to OGC/STA Server to OGC/STA Client\*\*

### 6.3.2 Configuration of the oneM2M CSE

#### 6.3.2.0 General Configuration Aspects

The IPE needs to perform configuration steps on the hosting CSE.

The IPE shall register itself as an Application Entity (AE) that is represented as an `<AE>` resource in a oneM2M resource tree.

The CSE uses notifications to communicate new events to the IPE (AE). Therefore, the `<AE>` resource shall have the \*requestReachability\* (rr) attribute set to 'true'.

The `<AE>` shall have a \*pointOfAccess\* (poa) attribute giving the protocol and address that the IPE is going to use to receive notifications from the CSE.

The message flow for the creation of an `<AE>` resource is shown in Figure 6.3.2.0-1:

1) The IPE requests to register an `<AE>` resource on the hosting CSE.

2) The hosting CSE evaluates the request, performs the appropriate checks, and registers the `<AE>` resource.

3) The hosting CSE responds with a successful result response upon successful creation of the `<AE>` resource; otherwise, it responds with an error.

![Figure 6.3.2.0-1: Message flow of an AE Creation in oneM2M](media/create\_ae\_flow.png)

\*\*Figure 6.3.2.0-1: Message flow of an AE Creation in oneM2M\*\*

#### 6.3.2.1 Communication direction oneM2M CSE towards IPE

It needs two `<container>` resources in the CSE for the operation of the IPE, one for outgoing data and one for incoming data.

The `<container>` resource that is appointed to hold the data to be forwarded to the OGC/STA side (outgoing data) has to be created, if not already existing.

The message flow for the creation of a `<container>` resource is shown in Figure 6.3.2.1-1:

1) The IPE sends a request to create a `<container>` resource.

2) The hosting CSE evaluates the request, performs the appropriate checks, and creates the `<container>` resource.

3) The hosting CSE responds with a successful result response upon successful creation of the `<container>` resource; otherwise, it responds with an error.

![Figure 6.3.2.1-1: Message flow of an Container Creation in oneM2M](media/create\_container\_flow.png)

\*\*Figure 6.3.2.1-1: Message flow of an Container Creation in oneM2M\*\*

A `<subscription>` resource shall be created under this `<container>` resource.

The `<subscription>` resource shall have the \*notificationURI\* attribute set to the \*resourceID\* of the `<AE>` resource.

The message flow for the creation of an `<subscription>` resource is shown in Figure 6.3.2.1-2:

1) The IPE sends a creation request for a `<subscription>` to the `<container>` that is appointed to hold the data to be forwarded to the OGC/STA side.

2) The hosting CSE evaluates the request and performs the appropriate checks and creates the `<subscription>` resource.

3) The hosting CSE responds with a successful result response of the `<subscription>` resource creation; otherwise, it responds with an error.

![Figure 6.3.2.1-2: Message flow of an Subcription Creation in oneM2M](media/create\_subscription\_flow1.png)

\*\*Figure 6.3.2.1-2: Message flow of an Subcription Creation in oneM2M\*\*

The CSE is now prepared to send data from oneM2M to OGC / STA via the IPE. As shown in Figure 6.3.2.1-3, a oneM2M Application Entity (AE), triggered by a sensor, sends data to the CSE by creating a `<contentInstance>` resource under the `<container>` resource that was appointed for outgoing data.

Since the IPE has subscribed to this `<container>` resource it receives a notification message along with all attributes of the `<contentInstance>` resource when new data arrives. The IPE maps the data from oneM2M to OGC / STA as described in 6.1 .

![Figure 6.3.2.1-3: Message flow from AE to CSE to IPE](media/config\_cse1.png)

\*\*Figure 6.3.2.1-3: Message flow from AE to CSE to IPE\*\*

#### 6.3.2.2 Communication direction IPE towards oneM2M CSE

The `<container>` resource that is appointed to hold the data from the OGC/STA side (incoming data) has to be created, if not already existing.

The message flow for the creation of a `<container>` resource is shown in Figure 6.3.2.1-1.

The CSE is now prepared to receive data from OGC / STA via the IPE. The IPE sends data as `<contentInstance>` resources to the dedicated `<container>` resource. If other oneM2M Application Entities are interested in this data, they may subscribe to the dedicated `<container>` resource. Alternatively, they can retrieve `<contentInstance>` resources from it in polling mode.

In Figure 6.3.2.2-1, the IPE (AE) sends data as `<contentInstance>` resources to the dedicated `<container>` resource. Subsequently, the AE receives a notification along with data contained in a `<contentInstance>` resource every time when the IPE creates new data.

![Figure 6.3.2.2-1: Data message flow from IPE to CSE to AE](media/config\_cse23.png)

\*\*Figure 6.3.2.2-1: Data message flow from IPE to CSE to AE\*\*

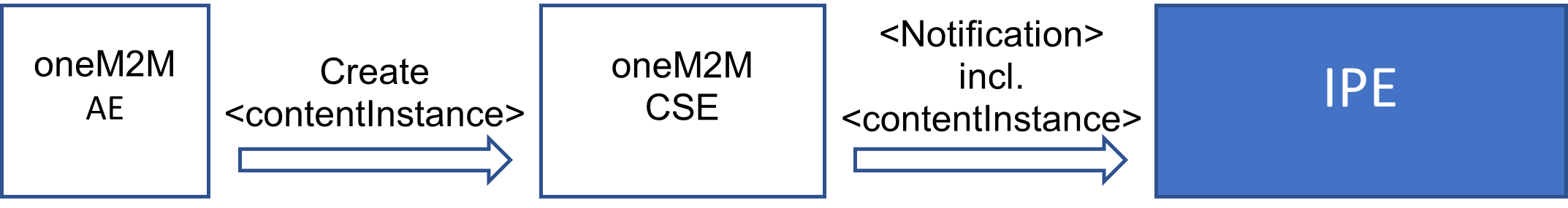
# Proforma copyright release text block

<mark>This text box shall immediately follow after the heading of an element (i.e. clause or annex) containing a proforma or template which is intended to be copied by the user. Such an element shall always start on a new page.</mark>

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

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

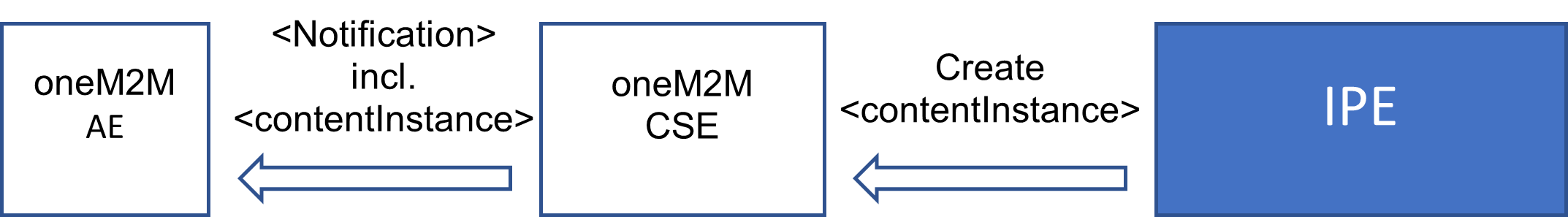
---/dev/null  
+++b/media/config\_cse1.png



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

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

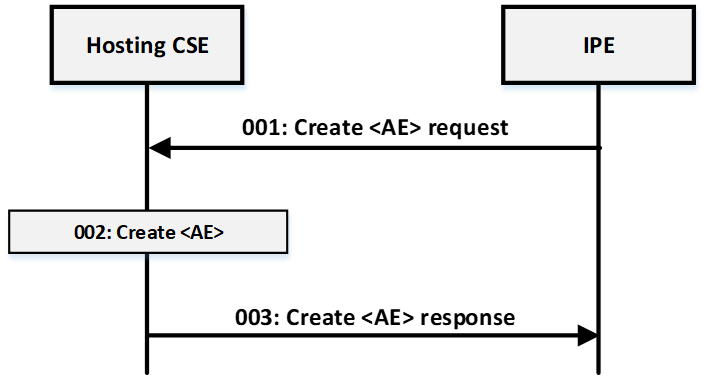
---/dev/null  
+++b/media/config\_cse23.png



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

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

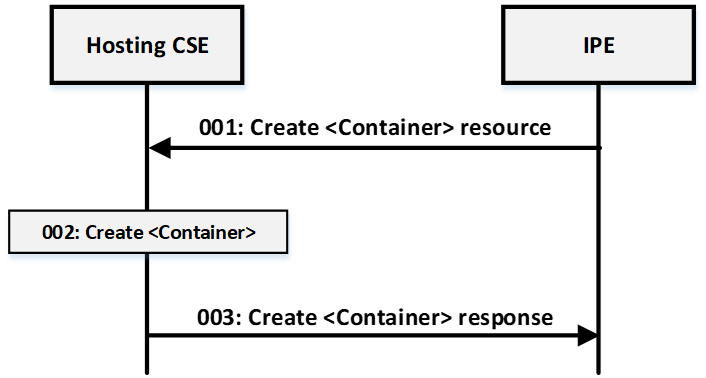
---/dev/null  
+++b/media/create\_ae\_flow.png



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

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

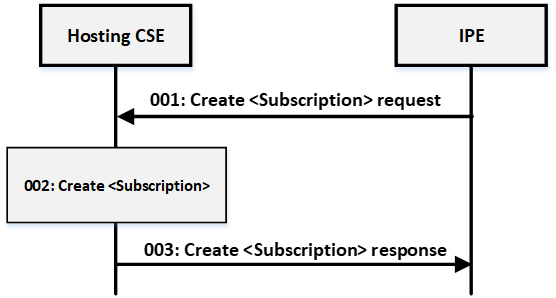
---/dev/null  
+++b/media/create\_container\_flow.png



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

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

---/dev/null  
+++b/media/create\_subscription\_flow1.png



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