|  |  |
| --- | --- |
| CHANGE REQUEST | |
| Meeting:\* | TP24 |
| Source:\* | InterDigital |
| Date:\* | 2016-07-20 |
| Contact:\* | Dale Seed ([dale.seed@interdital.com](mailto:dale.seed@interdital.com))  Bob Flynn ([bob.flynn@interdigital.com](mailto:bob.flynn@interdigital.com)) |
| Reason for Change/s:\* | Corrections identified during test purpose development |
| CR against: Release\* | Rel-2 |
| CR against: WI\* | Active <Work Item number>  MNT Maintenace / < Work Item number(optional)>  STE Small Technical Enhancements / < Work Item number (optional)>  Only ONE of the above shall be ticked |
| CR against: TS/TR\* | TS-0001-V2.9.0 |
| Clauses/Sub Clauses\* | 10.2.7.7 – 10.2.7.11 |
| 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 |
| Post Freeze checking:\* | This CR contains only essential changes and corrections? YES  NO  This CR is a mirror CR? YES  NO  if YES, please indicate the document number of the original CR:  ARC-2016-0348R01-fanoutPoint\_corrections |
| Template Version:23 February 2015 (Dot 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.

1. **Introduction**

This contribution document consist of editorial changes identified during test purposes development for group fanoutPoint requests.

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

#### 10.2.7.7 Create *<fanOutPoint>*

This procedure shall be used for creating the content of all members resources belonging to an existing *<group>* resource.

Table 10.2.7.7-1: *<fanOutPoint>* CREATE

|  |  |
| --- | --- |
| *<fanOutPoint>* CREATE | |
| Associated Reference Point | Mca, Mcc and Mcc' |
| Information in Request message | ***From:*** Identifier of the AE or the CSE that initiates the Request  ***To:*** The address of the *<fanOutPoint>* virtual resource  ***Content:*** The representation of the resource the Originator intends to create  ***Group Request Identifier:*** The group request identifier  ***Response Type:*** If the parameter is set to BlockingSynch, it indicates that the group hosting CSE shall return the aggregated response once. Otherwise if the parameter is set to nonBlockingRequestSynch, nonBlockingRequestAsynch or flexBlocking, it indicates that the Group Hosting CSE shall return the aggregated response in a batched mode  ***Result Expiration Time:*** Indicates the maximum time limit in which the Group Hosting CSE has to respond  ***Result Persistence:*** The Group Hosting CSE continues to update the response with any additional member responses recieved during that time |
| Processing at Originator before sending Request | The Originator shall request to create the resource that have the same content in all members resources belonging to an existing *<group>* resource by using a CREATE operation. The Request may address the virtual child resource *<fanOutPoint>* of the specific *<group>* resource of a group Hosting CSE. The request may also address the address that results from appending a relative address to the *<fanOutPoint>* address in order to create the resources that have the same content under the corresponding child resources represented by the relative address with respect to all members resources. The Originator may be an AE or CSE |
| Processing at Group Hosting CSE | For the CREATE procedure, the Group Hosting CSE shall:   * Check if the Originator has CREATE privilege in the *<accessControlPolicy>* resource referenced by the members *AccessControlPolicyIDs* in the *<group>* resource. In the case members *membersAccessControlPolicyIDs* is not provided the access control policy defined for the *<group>* resource shall be used * Upon successful validation, obtain the IDs of all members resources from the attribute *membersIDs* of the addressed *<group>* resource * Generate fan out requests addressing the obtained address (appended with the relative address if any) to the member hosting CSEs as indicated in figure 10.2.7.6-1.The ***From*** parameter in the fanout request is set to ID of the Originator from the request from the original Originator. The ***Response Type*** parameter in the fanout request may be set by the group hosting CSE differently according to its local policy * In the case that a member resource is a *<group>* resource and the request to be fanned out does not contain a group request identifier already, generate a unique group request identifier, include the group request identifier in all the requests to be fanned out and locally store the group request identifier * If the group Hosting CSE determines that multiple members resources belong to one CSE according to the IDs of the members resources, it may converge the requests accordingly before sending out. This may be accomplished by the group Hosting CSE creating a *<group>* resource on the members Hosting CSE to collect all the members on that members Hosting CSE * After receiving the responses from the members hosting CSEs, respond to the Originator with the aggregated results and the associated members list. Depending on the ***Response Type***, the Group Hosting CSE shall:   - **blockingRequest:** respond with the aggregated responses before the ***Result Expiration Time*** reaches and discard the member responses received after  - **nonBlockingRequestSynch:** prepare the *operationResult* of the <request> resource and indicate that if all the member responses have been aggregated by setting the *requestStatus* of the <request> resource before the ***Result Expiration Time*** reaches. Then within the time period of ***Result Persistence***, continue aggregating and updating the *operationResult*  - **nonBlockingRequestAsynch:** notify with the aggregated response from all or part of the members before the ***Result Expiration Time*** reaches. And within the time period of ***Result Persistance***, continue aggregate and notify with the remaining member responses if there are any  - **flexBlocking:** before the ***Result Expiration Time***, if all member responses has been aggregated, respond the aggregated response as in the blockingRequest case. Otherwise, response an acknowledgement together with the current aggregated member responses and the reference to the created <request> resource. Then within the time period of ***Result Persistence***, continue aggregate and deliver the remaining member response to the Originator as defined in the nonBlockingRequestSynch or the nonBlockingRequestAsynch case  (See note) |
| Processing at Member Hosting CSE | For the CREATE procedure, the Member Hosting CSE shall:   * Check if the request has a group request identifier. Check if the group request identifier is contained in the requested identifiers stored locally. If match is found, ignore the current request and respond an error. If no match is found, locally store the group request identifier until the expiration of the request expiration time or local policy * Check if the original Originator has the CREATE permission on the addressed resource. Upon successful validation, perform the create procedures for the corresponding type of addressed resource as described in other sub-clauses of clause 10.2 * Send the corresponding response to the Group Hosting CSE |
| Information in Response message | Converged responses from members hosting CSEs |
| Processing at Originator after receiving Response | None |
| Exceptions | * Same request with identical group request identifier received * Originator does not have the CREATE permission to access the *<fanOutPoint>* resource |
| NOTE: If ***Result Expiration Time*** or ***Result Persistance*** is not provide in the original request from the Originator, the group hosting CSE may decide the timer based on its local policy. | |

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

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

#### 10.2.7.8 Retrieve *<fanOutPoint>*

This procedure shall be used for retrieving the content of all member resources belonging to an existing *<group>* resource.

Table 10.2.7.8-1: *<fanOutPoint>* RETRIEVE

|  |  |
| --- | --- |
| *<fanOutPoint>* RETRIEVE | |
| Associated Reference Point | Mca, Mcc and Mcc' |
| Information in Request message | ***From:*** Identifier of the AE or the CSE that initiates the Request  ***To:*** The address of the *<fanOutPoint>* virtual resource  ***Content:*** The representation of the resource the Originator intends to retrieve  ***Group Request Identifier:*** The group request identifier  ***Response Type:*** If the parameter is set to BlockingSynch, it indicates that the group hosting CSE shall return the aggregated response once. Otherwise if the parameter is set to nonBlockingRequestSynch or nonBlockingRequestAsynch, it indicates that the Group Hosting CSE shall return the aggregated response in a batched mode.  ***Result Expiration Time:*** Indicates the maximum time limit in which the Group Hosting CSE has to respond  ***Result Persistence:*** The Group Hosting CSE continues to update the response with any additional member responses recieved during that time |
| Processing at Originator before sending Request | The Originator shall request to obtain the resource or specific attributes of all member resources belonging to an existing *<group>* resource by using a RETRIEVE operation. The request may address the virtual child resource *<fanOutPoint>* of the specific *<group>* resource of a group Hosting CSE. The request may also address the address that results from appending a relative address to the *<fanOutPoint>* address in order to retrieve the corresponding attributes or child resources represented by the relative address with respect to all members resources. The Originator may be an AE or CSE |
| Processing at Group Hosting CSE | For the RETRIEVE procedure, the Group Hosting CSE shall:   * Check if the Originator has RETRIEVE permission in the *<accessControlPolicy>* resource referenced by the *membersAccessControlPolicyIDs* in the addressed *<group>* resource. In the case *membersAccessControlPolicyIDs* is not provided, the access control policy defined for the group resource shall be used * Upon successful validation, obtain the IDs of all members resources from the *membersIDs* attribute of the addressed *<group>* resource * Generate fan out requests addressing the obtained address (appended with the relative address if any) to the members hosting CSEs as indicated in figure 10.2.7.6-1.The ***From*** parameter in the fanout request is set to ID of the Originator from the request from the original Originator. The ***Response Type*** parameter in the fanout request may be set by the group hosting CSE differently according to its local policy * In the case that a member resource is a *<group>* resource, generate a unique group request identifier and the request to be fanned out does not contain a group request identifier already, include the group request identifier in all the requests to be fanned out and locally store the group request identifier * If the group hosting CSE determines that multiple members resources belong to one CSE according to the IDs of the members resources, it may converge the requests accordingly before sending out. This may be accomplished by the group Hosting CSE creating a *<group>* resource on the members Hosting CSE to collect all the members on that members Hosting CSE * After receiving the responses from the members hosting CSEs, respond to the Originator with the aggregated results and the associated members list. Depending on the ***Response Type***, the Group Hosting CSE shall:   - BlockingRequest: respond with the aggregated responses before the ***Result Expiration Time*** reaches and discard the member responses received after.  - nonBlockingRequestSynch: prepare the *operationResult* of the <request> resource and indicate that if all the member responses have been aggregated by setting the *requestStatus* of the <request> resource before the ***Result Expiration Time*** reaches. Then within the time period of ***Result Persistence***, continue aggregating and updating the *operationResult*.  - nonBlockingRequestAsynch: notify with the aggregated response from all or part of the members before the ***Result Expiration Time*** reaches. And within the time period of ***Result Persistance***, continue aggregate and notify with the remaining member responses if there are any.  - flexBlocking: before the ***Result Expiration Time***, if all member responses has been aggregated, respond the aggregated response as in the blockingRequest case. Otherwise, response an acknowledgement together with the current aggregated member responses and the reference to the created <request> resource. Then within the time period of ***Result Persistence***, continue aggregate and deliver the remaining member response to the Originator as defined in the nonBlockingRequestSynch or the nonBlockingRequestAsynch case.  (See note) |
| Processing at Member Hosting CSE | For the RETRIEVE procedure, the Member Hosting CSE shall:   * Check if the request has a group request identifier. Check if the group request identifier is contained in the requested identifier stored locally. If match is found, ignore the current request and respond an error. If no match is found, locally store the request identifier until the expiration of the request expiration time or local policy * Check if the original Originator has the RETRIEVE permission on the addressed resource. Upon successful validation, perform the retrieve procedures for the corresponding type of addressed resource as described in other sub-clauses of clause 10.2 * Send the corresponding response to the group Hosting CSE |
| Information in Response message | Converged responses from members hosting CSEs |
| Processing at Originator after receiving Response | None |
| Exceptions | * Same request with identical group request identifier received * Originator does not have RETRIEVE permission to access the *<fanOutPoint>* resource |
| NOTE: If ***Result Expiration Time*** or ***Result Persistance*** is not provide in the original request from the Originator, the group hosting CSE may decide the timer based on its local policy. | |

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

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

#### 10.2.7.9 Update *<fanOutPoint>*

This procedure shall be used for updating the content of all member resources belonging to an existing *<group>* resource.

Table 10.2.7.9-1: *<fanOutPoint>* UPDATE

|  |  |
| --- | --- |
| *<fanOutPoint>* UPDATE | |
| Associated Reference Point | Mca, Mcc and Mcc' |
| Information in Request message | ***From:*** Identifier of the AE or the CSE that initiates the Request  ***To:*** The address of the *<group>* resource  ***Content:*** The representation of the resource the Originator intend to Update  ***Group Request Identifier:*** The group request identifier  ***Response Type:*** If the parameter is set to BlockingSynch, it indicates that the group hosting CSE shall return the aggregated response once. Otherwise if the parameter is set to nonBlockingRequestSynch or nonBlockingRequestAsynch, it indicates that the Group Hosting CSE shall return the aggregated response in a batched mode  ***Result Expiration Time:*** Indicates the maximum time limit in which the Group Hosting CSE has to respond  ***Result Persistence:*** The Group Hosting CSE continues to update the response with any additional member responses recieved during that time |
| Processing at Originator before sending Request | The Originator shall request to update all member resources belonging to an existing *<group>* resource with the same data by using a UPDATE operation. The request may address the virtual child resource *<fanOutPoint>* of the specific *<group>* resource of a group Hosting CSE. The request may also address the address that results from appending a relative address to the *<fanOutPoint>* in order to update the corresponding child resources represented by the relative address with respect to all *<members>* resources. The Originator may be an AE or CSE |
| Processing at Group Hosting CSE | For the UPDATE procedure, the Group Hosting CSE shall:   * Check if the Originator has UPDATE permission in the *<accessControlPolicy>* resource referenced by the *membersAccessControlPolicyIDs* in the group resource. In the case members *membersAccessControlPolicyIDs* is not provided the access control policy defined for the group resource shall be used * Upon successful validation, obtain the IDs of all member resources from the attribute *membersIDs* of the addressed *<group>* resource * Generate fan out requests addressing the obtained address (appended with the relative address if any) to the members hosting CSEs as indicated in figure10.2.7.6-1.The ***From*** parameter in the fanout request is set to ID of the Originator from the request from the original Originator. The ***Response Type*** parameter in the fanout request may be set by the group hosting CSE differently according to its local policy * In the case that a member resource is a *<group>* resource and the request to be fanned out does not contain a group request identifier already, generate a unique group request identifier, include it in all the requests to be fanned out and locally store the group request identifier * If the group Hosting CSE determines that multiple members resources belong to one CSE according to the IDs of the member resources, it may converge the requests accordingly before sending out. This may be accomplished by the group Hosting CSE creating a *<group>* resource on the member Hosting CSE to collect all the members on that members Hosting CSE * After receiving the responses from the members hosting CSEs, respond to the Originator with the aggregated results and the associated members list. Depending on the ***Response Type***, the Group Hosting CSE shall:   - BlockingRequest: respond with the aggregated responses before the ***Result Expiration Time*** reaches and discard the member responses received after  - nonBlockingRequestSynch: prepare the *operationResult* of the <request> resource and indicate that if all the member responses have been aggregated by setting the *requestStatus* of the <request> resource before the ***Result Expiration Time*** reaches. Then within the time period of ***Result Persistence***, continue aggregating and updating the *operationResult*  - nonBlockingRequestAsynch: notify with the aggregated response from all or part of the members before the ***Result Expiration Time*** reaches. And within the time period of ***Result Persistance***, continue aggregate and notify with the remaining member responses if there are any  - flexBlocking: before the ***Result Expiration Time***, if all member responses has been aggregated, respond the aggregated response as in the blockingRequest case. Otherwise, response an acknowledgement together with the current aggregated member responses and the reference to the created <request> resource. Then within the time period of ***Result Persistence***, continue aggregate and deliver the remaining member response to the Originator defined in the nonBlockingRequestSynch or the nonBlockingRequestAsynch case  (See note) |
| Processing at Member Hosting CSE | For the UPDATE procedure, the Member Hosting CSE shall:   * Check if the request has a group request identifier. Check if the request identifier is contained in the requested identifier stored locally. If match is found, ignore the current request and respond an error. If no match is found, locally store the request identifier until the expiration of the request expiration time or local policy * Check if the original Originator has the UPDATE permission on the addressed resource. Upon successful validation, perform the update procedures for the corresponding type of addressed resource as described in other sub-clauses of clause 10.2 * Send the corresponding response to the group Hosting CSE |
| Information in Response message | Converged responses from members hosting CSEs |
| Processing at Originator after receiving Response | None |
| Exceptions | * Same request with identical group request identifier received * Originator does not have the UPDATE permissions to access the *<fanOutPoint>* resource |
| NOTE: If ***Result Expiration Time*** or ***Result Persistance*** is not provide in the original request from the Originator, the group hosting CSE may decide the timer based on its local policy. | |

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

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

#### 10.2.7.10 Delete *<fanOutPoint>*

This procedure shall be used for deleting the content of all members resources belonging to an existing *<group>* resource.

Table 10.2.7.10-1: *<fanOutPoint>* DELETE

|  |  |
| --- | --- |
| *<fanOutPoint>* DELETE | |
| Associated Reference Point | Mca, Mcc and Mcc' |
| Information in Request message | ***From:*** Identifier of the AE or the CSE that initiates the Request  ***To:*** The address of the *<fanOutPoint>* virtual resource  ***Content:*** The representation of the resource the Originator intends to delete  ***Group Request Identifier:*** The group request identifier  ***Response Type:*** If the parameter is set to BlockingSynch, it indicates that the group hosting CSE shall return the aggregated response once. Otherwise if the parameter is set to nonBlockingRequestSynch or nonBlockingRequestAsynch, it indicates that the Group Hosting CSE shall return the aggregated response in a batched mode  ***Result Expiration Time:*** Indicates the maximum time limit in which the Group Hosting CSE has to respond  ***Result Persistence:*** The Group Hosting CSE continues to update the response with any additional member responses recieved during that time |
| Processing at Originator before sending Request | The Originator shall request to delete all members resources belonging to an existing *<gro*u*p>* resource by using a DELETE operation. The request may address the virtual child resource *<fanOutPoint>* of the specific *<group>* resource of a group Hosting CSE. The request may also address the address that results from appending a relative address to the *<fanOutPoint>* in order to delete the corresponding child resources represented by the relative address with respect to all member resources. The Originator may be an AE or a CSE |
| Processing at Group Hosting CSE | For the DELETE procedure, the *<group>* Hosting CSE shall:   * Check if the Originator has DELETE permission in the *<accessControlPolicy>* resource referenced by the *membersAccessControlPoliciIDs* in the *<group>* resource. In the case *membersAccessControlPolicyIDs* is not provided the access control policy defined for the group resource shall be used * Upon successful validation, obtain the IDs of all member resources from the attribute *membersIDs* of the addressed *<group>* resource * Generate fan out requests addressing the obtained address (appended with the relative address if any) to the member hosting CSEs as indicated in figure 10.2.7.6-1. ***From*** parameter in the fanout request is set to ID of the Originator from the request from the original Originator. The ***Response Type*** parameter in the fanout request may be set by the group hosting CSE differently according to its local policy * In the case that the members resources is a *<group>* resource and the request to be fanned out does not contain a group request identifier already, generate a unique group request identifier, include the group request identifier in all the requests to be fanned out and locally store the group request identifier * If the *<group>* Hosting CSE determines that multiple members resources belong to one CSE according to the IDs of the members resources, it may converge the requests accordingly before sending out. This may be accomplished by the group Hosting CSE creating a *<group>* resource on the member Hosting CSE to collect all the members on that member Hosting CSE * After receiving the responses from the members hosting CSEs, respond to the Originator with the aggregated results and the associated members list. Depending on the ***Response Type***, the Group Hosting CSE shall:   - BlockingRequest: respond with the aggregated responses before the ***Result Expiration Time*** reaches and discard the member responses received after  - nonBlockingRequestSynch: prepare the *operationResult* of the <request> resource and indicate that if all the member responses have been aggregated by setting the *requestStatus* of the <request> resource before the ***Result Expiration Time*** reaches. Then within the time period of ***Result Persistence***, continue aggregating and updating the *operationResult*  - nonBlockingRequestAsynch: notify with the aggregated response from all or part of the members before the ***Result Expiration Time*** reaches. And within the time period of ***Result Persistance***, continue aggregate and notify with the remaining member responses if there are any  - flexBlocking: before the ***Result Expiration Time***, if all member responses has been aggregated, respond the aggregated response as in the blockingRequest case. Otherwise, response an acknowledgement together with the current aggregated member responses and the reference to the created <request> resource. Then within the time period of ***Result Persistence***, continue aggregate and deliver the remaining member response to the Originator as defined in the nonBlockingRequestSynch or the nonBlockingRequestAsynch case  (See note) |
| Processing at Member Hosting CSE | For the DELETE procedure, the Members Hosting CSE shall:   * Check if the request has a group request identifier. Check if the group request identifier is contained in the requested identifier stored locally. If match is found, ignore the current request and respond an error. If no match is found, locally store the group request identifier until the expiration of the request expiration time or local policy * Check if the original Originator has the DELETE permission on the addressed resource. Upon successful validation, perform the delete procedures for the corresponding type of addressed resource as described in other sub-clauses of clause 10.2 * Send the corresponding response to the Group Hosting CSE |
| Information in Response message | Converged responses from members hosting CSEs |
| Processing at Originator after receiving Response | None |
| Exceptions | * Same request with identical group request identifier received * Originator does not have the DELETE permissions to access the *<fanOutPoint>* resource |
| NOTE: If ***Result Expiration Time*** or ***Result Persistance*** is not provide in the original request from the Originator, the group hosting CSE may decide the timer based on its local policy. | |

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

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

#### 10.2.7.11 Subscribe and Un-Subscribe *<fanOutPoint>* of a group

This procedure shall be used for receiving information about modifications of all member resources belonging to an existing *<group>* resource.

Table 10.2.7.11-1: *<fanOutPoint>* Subscribe/Un-subscribe

|  |  |
| --- | --- |
| *<fanOutPoint>* Subscribe/Un-subscribe | |
| Associated Reference Point | Mca, Mcc and Mcc' |
| Information in Request message | ***From:*** Identifier of the AE or CSE that initiates the request  ***To:*** The address of the <fanOutPoint> resource appended with the ID of the *<subscription>* resource to be created  ***Group Request Identifier:*** The group request identifier |
| Processing at Originator before sending Request | The Originator shall request to create a subscription resource under all member resources belonging to an existing *<group>* resource by using a CREATE operation. The request may address the virtual  child resource *<fanOutPoint>* of the specific *<group>* resource of a group Hosting  CSE. The request may also address the address that results from appending a relative  address to the *<fanOutPoint>* in order to create the corresponding subscription to the resource represented by the relative address with respect to all member resources. In both cases the targeted resource shall the parent of the newly created <subscription> resource(s). The request shall include *notificationForwardingURI* attribute if the Originator wants the group Hosting CSE to aggregate the notifications. The request shall include the required information and may include the optional information as described in subscription management clause 10.2.11. The Originator may be an AE or a CSE |
| Processing at Group Hosting CSE | The *<group>* Hosting CSE shall:   * Check if the Originator has CREATE privilege in the *<accessControlPolicy>* resource referenced by the *membersAccessControlPolicyIDs* in the group resource. In the case *membersAccessControlPolicyIDs* is not provided the access control policy defined for the group resource shall be used * If the subscription resource in the request contains an *notificationForwardingURI* attribute, assign a URI to replace the *notificationURI* of the subscription resource which will be used to receive notifications from member hosting CSEs. The ID of the *<group>* resource shall be set to the *groupID* attribute of the *<subscription>* resource. The group Hosting CSE shall maintain the mapping of the generated *notificationURI* and the former *notificationURI* * Upon successful validation, obtain the IDs of all member resources from the   attribute membersIDs of the addressed <group> resource   * Generate fan out requests addressing the obtained address (appended with   the relative address if any) to the member hosting CSEs as indicated in figure  10.2.7.6-1. From parameter in the request is set to ID of the Originator from  the request from the original Originator   * If the group Hosting CSE determines that multiple members resources belong to one CSE according to the IDs of the member resources, it may converge the requests accordingly before sending out. This may be accomplished by the *<group>* Hosting CSE creating a *<group>* resource on the members Hosting CSE to collect all the members on that members Hosting CSE * After receiving the responses from the member hosting CSEs, respond to the Originator with the aggregated results and the associated members list |
| Processing at Member Hosting CSE | For the subscribe/un-subscribe procedure, the Members Hosting CSE shall treat the request received from the group Hosting CSE as a normal SUBSCRIBE request on the addressed member resource as if it comes from the original Originator. Therefore the members Hosting CSE shall:   * Check if the request has a group request identifier. Check if the group request   identifier is contained in the requested identifier stored locally. If match is  found, ignore the current request and respond an error. If no match is found,  locally store the group request identifier until the expiration of the request expiration time or local policy   * Check if the original Originator has the READ permission on the members resource * Upon successful validation, perform the subscribe procedures for the corresponding type of member resource as described in clause 10.2.12 * Send the corresponding response to the group Hosting CSE |
| Information in Response message | Converged responses from member hosting CSEs |
| Processing at Originator after receiving Response | None |
| Exceptions | * Same request with identical request identifier received * Originator does not have the access control privilege to access the *<fanOutPoint>* resource |

Un-subscribing to the members of a <group> resource uses the “Delete <fanoutPoint>” procedure defined in 10.2.7.10.

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